

マイコン技術カルブックス MICOM TECHNICAL BOOKS

# PC-8801mkIIsR /FR/MR 活用研究

FH/MHシリーズ対応

マシン語プログラミングからFM音源・  
グラフィックテクニックまで!

呉 英二/石田和久 共著













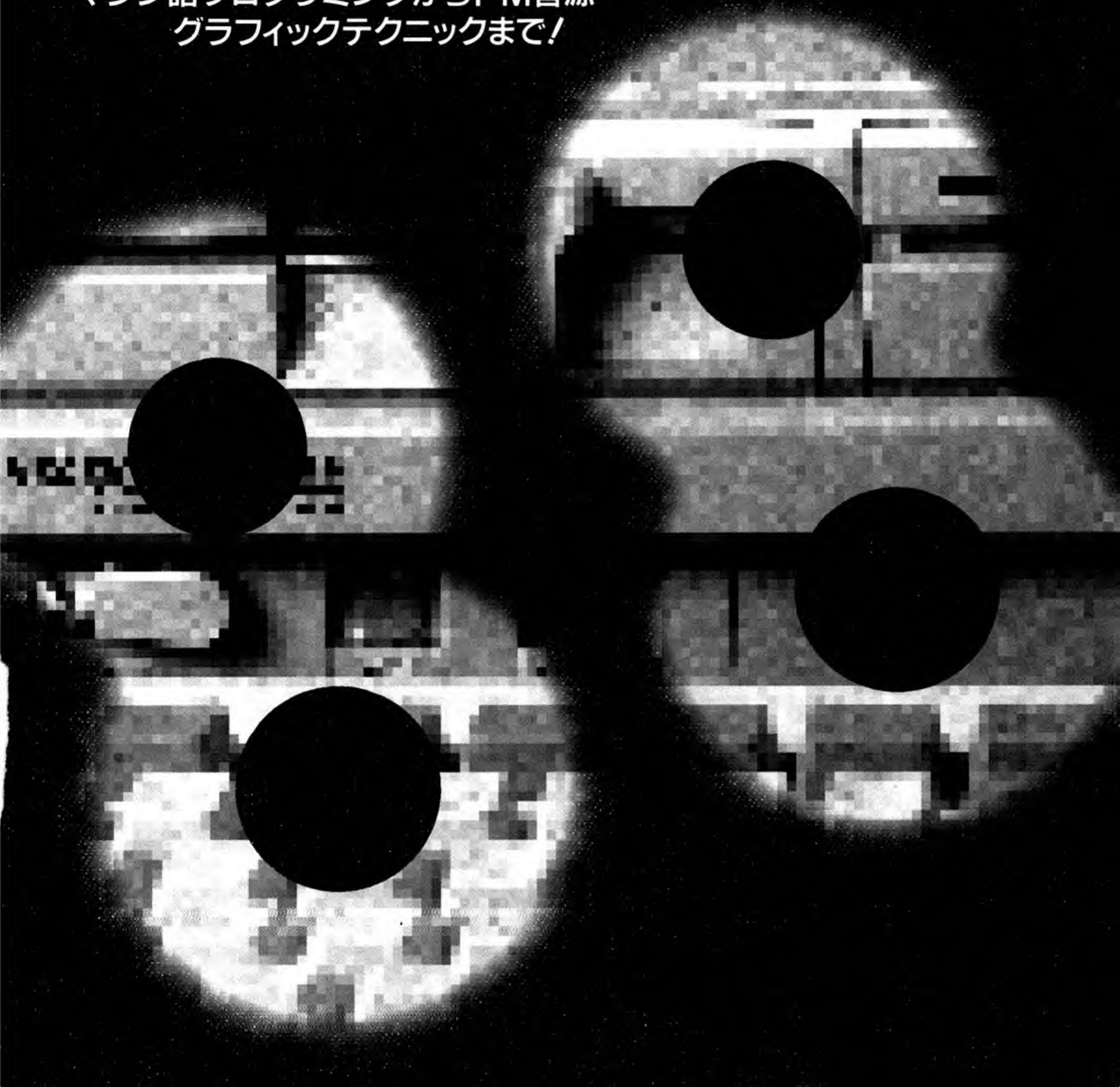


# PC-8801mkIISR

## /FR/mr 活用研究

FH/MHシリーズ対応

マシン語プログラミングからFM音源・  
グラフィックテクニックまで!









## はじめに

初期のパーソナルコンピュータでは、表示能力が小さく、決まった文字や形を表示する事しかできませんでした。反面、ゲームを作る時にはこの様な制限があるため、かえって作りやすかったものです。

ちょっとしたアイデアさえあれば、BASICで十分遊べるものも作れました。

ところが、コンピュータの表現力が増すにつれ、きれいなグラフィックを使えるようになったのはよいのですが、それを動かしてプログラムを作るとなると、BASICでは遅くて使いものになりません。

そこで、マシン語を使う必要に迫られるのですが、マシン語なんてとても使えそうにないとあきらめてしまいがちです。

事実、マシン語でプログラムを組むのは骨がおれます。しかし、決して難しいものではありません。面倒さはあるますが、極めて単純なものなのです。

私も、モニタで打ち込んだ短いマシン語プログラムが走り、画面にもキャラクターが表示されるまでは、自分にマシン語プログラムなどできそうにないと思っていました。

でも、できる事さえ分かってしまえば、後は小さいプログラムの積み重ねです。

本書はマシン語の世界に飛び込もうとするあなたのために、そのきっかけとなればと思い書きおろしました。そのため、具体的なプログラムを多く例に上げました。“どうやって？”と考えるよりもあなたのコンピュータにプログラムを打ち込み、実験してみましょう。

マシン語を始めるのに、“アセンブラー”などの特別な道具はいりません。あなたのPC-88と“やる気”さえあればよいのです。素手でマシン語を牛耳ってやりましょう。

一つでもきっかけがつかめれば、プログラム作りがいかにも面白いものか発見できると思います。その時からあなたもマシン語プログラマーですよ。

最後に、お世話になりました電波新聞社出版部の皆様、NECの皆様、ナニワさん、塚越さんに感謝いたします。

1987年2月2日

呉 英二

石田和久



## PC-8801mkIISR/FR/MR活用研究

### 第1章

#### 88シリーズの魅力を探る

7

1-1 88シリーズノ SR以降の特徴	8
高速グラフィック処理	8
FM音源付きPSG	9
1-2 88シリーズの歴史	9
PC-8801	10
PC-8801mkII	10
PC-8801mkIISR	11
PC-8801mkIITR	11
PC-8801mkIIFR/MR	12
PC-8801FH/MH	13

### 第2章

#### I/Oの実戦的活用

15

2-1 キースキャン	16
2-2 画面コントロール	19
キャラクタを作る	19
キャラクタをデータにする	23
キャラクタを表示する	24
キャラクタを消す	41
2-3 パレット・コントロール	44
2-4 フロッピーディスクコントロール	46

### 第3章

#### BGMプログラミング

51

3-1 割り込みとは?	52
PC-88はマルチCPU?	52
割り込みってなに	52
PC-8801に割り込みをかけるには?	53
みんなで割り込みを使おう!	54
PC-8801の割り込みに関するハード	56
割り込みテーブルと、レジスタ	58
まとめ	59

#### 3-2 FM音源活用法

いよいよFM音源!	62
FM音源の使い方	62
マシン語による音出し	64
マシン語による音色設定	65
OPNのイニシャライズ	76
FM音源の音色設定	76
まとめ	78

#### 3-3 FM音源BGMルーチン

集大成ノFM音源BGMルーチン	81
最後のプレゼント	97
最後に	99

### 第4章

#### グラフィック・ハイ・テク

#### ニック

101

#### 4-1 グラフィックデータの圧縮・展開

グラフィックデータの保存方法	102
圧縮の基本的な考え方	103
グラフィックに使われるデータ	104
圧縮へのいくつかのチャレンジ	105
圧縮の具体的な方法	105

#### 4-2 高速DISK LOADER

高速DISK LOADERとは?	111
DOSとは何じゃ?	111
PC-8801のディスクのハード構成	111
何が高速か?	113
メインルーチンの説明	116
サンプルプログラム & 使い方	118
みんなで使おう高速DOS!	120

### 第5章

#### Z80マシン語入門

123

#### 5-1 はじめに

#### 5-2 コンピュータの成り立ち

#### 5-3 マシン語の書き方

#### 5-4 レジスタって何だろう?



5-5 “LD”命令は基本の基本	129
スタックポインタとは	133
PUSH・POP使用上の注意	137

5-6 “旗(フラグ)”の事	137
ゼロ(Z)フラグ	138
キャリー(CY)フラグ	138

5-7 計算させる方法	139
特別な演算	143

5-8 GOTO(ジャンプ)してみる	146
無条件絶対ジャンプ	147
ゼロフラグの内容によるジャンプ	147
キャリーフラグの内容によるジャンプ	149
特殊なジャンプ	153

5-9 GOSUBは知っていますね?	154
--------------------	-----

5-10 ブロック転送は速いのなんのって	157
----------------------	-----

5-11 かけ算ができればいいなあ	159
RLC(RRC)命令	159
RLA(RRA)命令	159
2倍以上のかけ算	159
ランダムテクニック	161

5-12 周辺装置とお話する	163
IN命令	163
OUT命令	164

5-13 その他の命令はあまり気にせず	164
---------------------	-----

## 第6章 マシン語ゲームを作ろう -----167

6-1 まず、どんなゲームを作るか決める	168
----------------------	-----

6-2 画面構成やフィールドを考える	169
--------------------	-----

6-3 ワークエリアを作りながら動かし方を考える	170
--------------------------	-----

6-4 フィールドを作り、キャラクタを動かす	171
動かすために必要なサブルーチンを準備する	171
動き回るフィールドを作る	176
“判定”する	180
キー入力で自分のキャラクタを動かす	186
データで移動させる	191
追いかけて移動する	198

6-5 スクロール画面を作るには	211
------------------	-----

6-6 まとめ	214
---------	-----

## 第7章 GAMMA5コンストラク ションゲーム -----217

7-1 ゲームの内容	218
登場キャラクタ	218
アイテム	220
移動キャラクタ	220

7-2 プログラム入力方法	220
---------------	-----

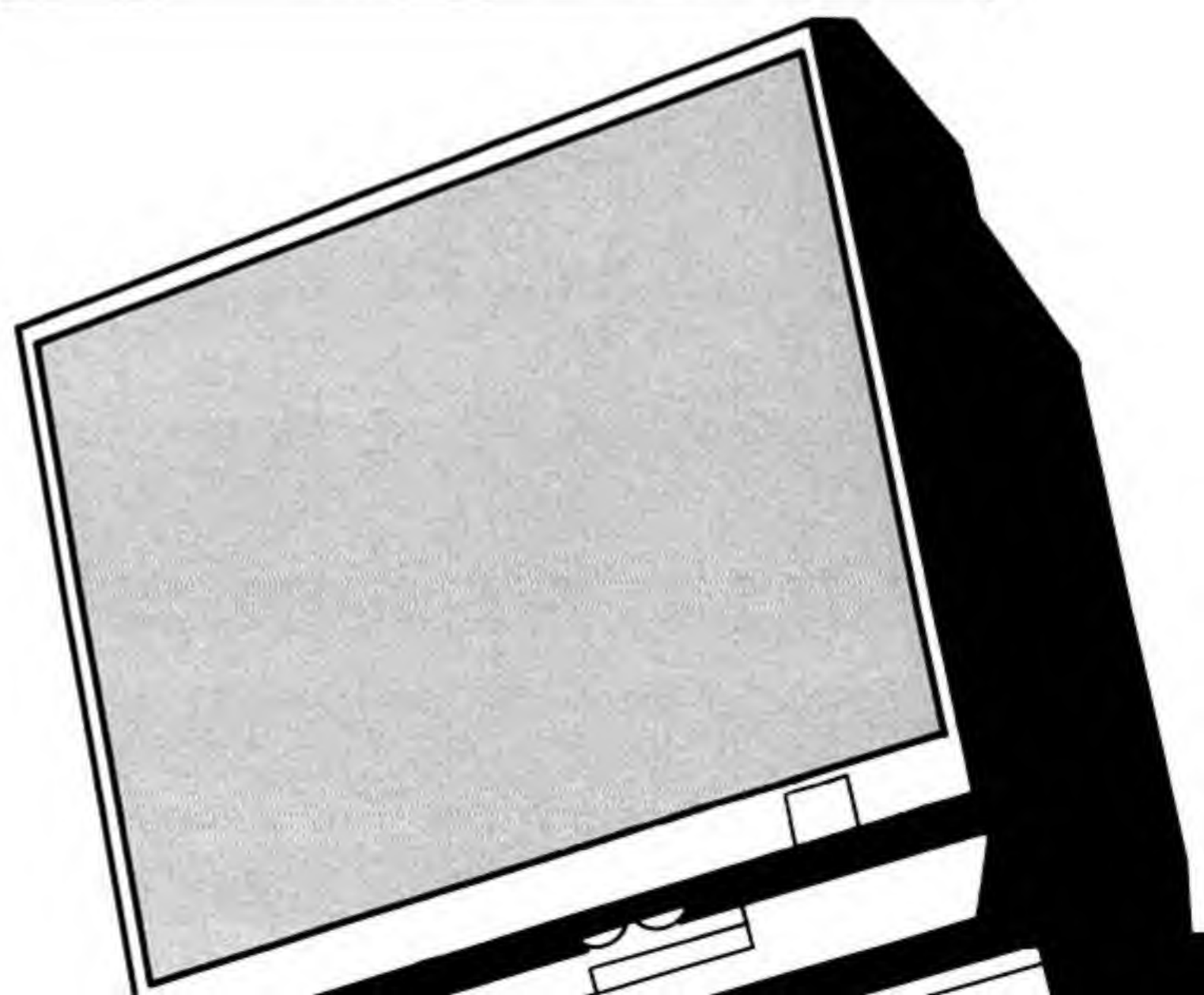
7-3 マップエディタ	224
Mコマンド	225
Gコマンド	225
Sコマンド	226
Tコマンド	226

7-4 敵攻撃データ	231
攻撃タイプ	232
防御タイプ	232

7-5 プログラムリスト	234
--------------	-----

## 付 録

A $\mu$ COM-82インストラクション・セット	258
B $\mu$ COM-82マシン語 $\leftrightarrow$ ニーモニック対応表	267
C $\mu$ COM-82ニーモニック $\leftrightarrow$ マシン語対照表	270
D 10進 $\leftrightarrow$ 16進変換表	274
E 1バイト符号付16進数	275









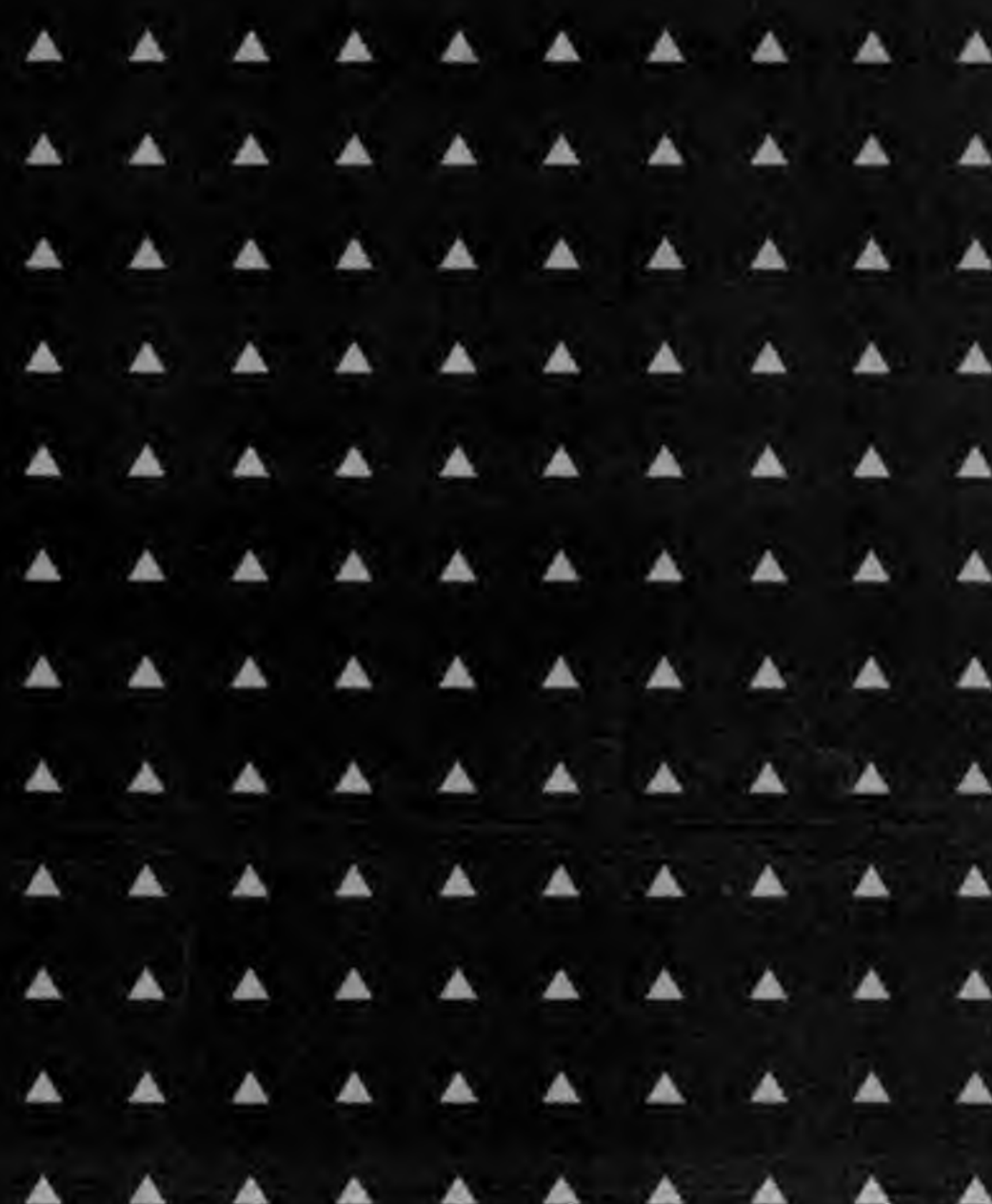
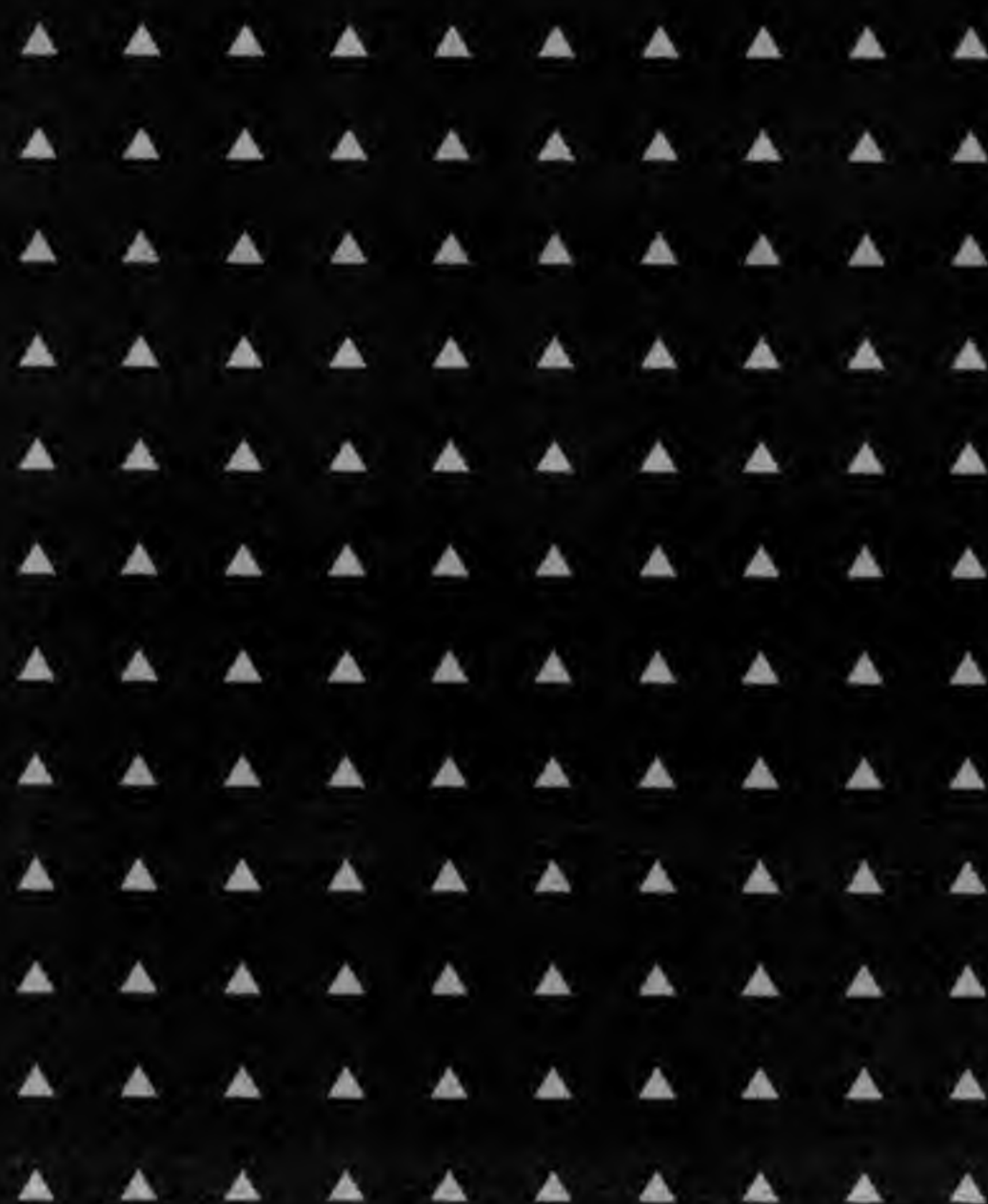
第 1 章

1

# 88シリーズの魅力を探る

1-1 88シリーズ / SR以降の特徴

1-2 88シリーズの歴史





# 第1章

## 88シリーズの魅力を探る

### 1-1 88シリーズ/ SR以降の特徴

最近の市販ソフト、特にゲームでは、PC-8801mkII SR以降の機種専用のものが目立つようになってきました。ソフトを作る側からすると、できるだけ多くのユーザーに楽しんでいただくため、多くのハード(機種)に対応したものを作りたいのはやまやまです。

しかし、より良いゲームのためにはより良いグラフィック表現と、より良い音楽が要求されるようになると、それなりの機能を備えた機種に限定されざるをえません。

すなわち、PC-8801mkII SR以降と、それより前の機種との大きな差は、高速グラフィックとFM音源を使ったミュージック機能にあると言ってよいでしょう。

本書では、これらの使い方をできるだけ具体的に解説してありますが、大まかにその機能を説明しましょう。

### その1 高速グラフィック処理

#### a) 全体処理のスピードアップ

SR以前の機種ではテキスト画面表示をするために、いちいちCPUの処理をさまたげていました。ですから、テキスト画面を表示させないようにすると、2、3割はグラフィック表示速度が上ったものです。しかし、SR以降では処理のタイミングが非常に効率よくなっており、テキストを表示させるさせないにかかわらず、マシンの基本性能である4MHzのクロックに従った動作速度を持ちます。

SR以前では、いくら同じ4MHzのクロックを使っているとは言え、表示などにウェイトが入りますから、見かけ上は4MHzで動作していないように見えるのです。

同様に、FH/MHではクロックが8MHzになりましたが、全体の処理速度がFR/MRなどと較べて、2倍の速度で動くわけではありません。ディスクドライブコントローラーや、FM音源は4MHzのままですし、他のメモリも高いクロックにはついて行けないものもありますので、ある程度ウェイトが必要になってきます。そのため実質では、SR以降の機種に較べて、1.5倍程の速さになっています。

#### b) "ALU" と呼ばれる強力な機能がっている

SR以前のグラフィックRAMへのアクセスは、青、赤、緑の3プレーン別々にバンク切り替えしてセットしていました。例えば、画面のある所に白い点を打とうとすると、まず、青のバンクを指定して点のデータをプットします。しかし、ただプットしたのではその点のまわりの点をこわしてしまいますから、画面のデータを読んで、書き込みたいデータと重ね合わせ、グラフィックRAMに入れます。



これを残りの赤と緑についても同様に行わなければなりません。ただ点をうつと言っても簡単ではなかったのです。

しかし、SR以降の機種では3プレーン同時にデータをセットする事ができ、しかも自動的に重ね合わせもやってくれるので、作業が簡単で高速になりました。白い点をうちたければ、モードを設定して点のデータを一度書き込むだけでよいのです。

BASICで"CLS 2"のオールクリアを実行してみると、その速さの違いがわかります。

ALUにはまだいくつかのモードがありますが、グラフィックの高速化に関しては画期的な機能と言ってよいでしょう。

## その2 FM音源付きPSG

SR以前では決まった周波数を発する単純な音源しかなく、音階を出すためにはかなりの工夫が必要でした。しかし、SR以降では、ごく簡単に音階、音量を指定できる3音と1ノイズがつき、さらに様々な音色を表現できるFM音源が3音ついています。

今やパーソナルコンピュータのホビーユースには欠かせなくなったBGMも、これらの音源により高度な表現が可能となりました。

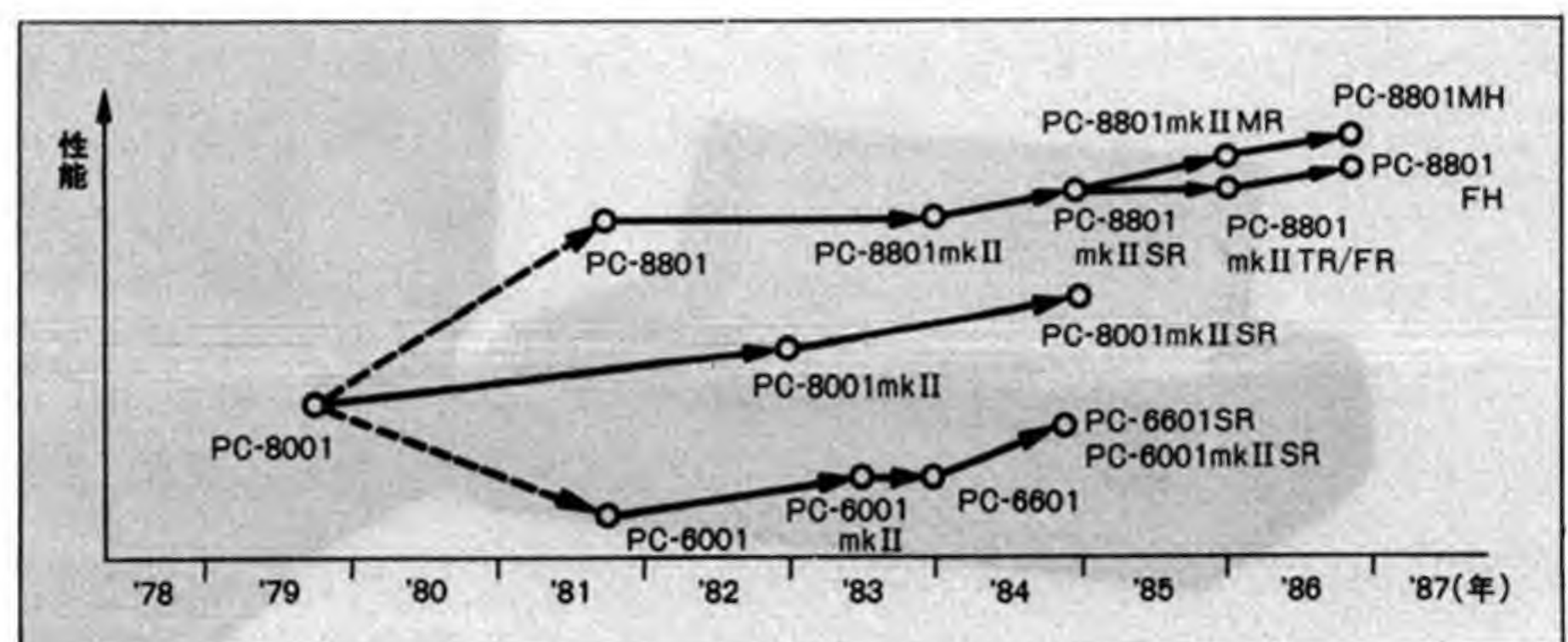
しかしながら、FM音源の方は仕組みが複雑ですので、ユーザーのイメージした音を自由に作り出すのは中々大変の様です。でも、62種のサンプルをROMでサポートしており、実用にはこのデータを活用すれば充分なようです。BASICのPLAY文を試された方は、これらの音色をお聞きになっている事でしょう。

本書では、BASICの壁をやぶって、FM音源によるBGMの具体的な使用にチャレンジしています。

と、まあこの2点を強調しましたが、この他にも、漢字ROMの標準装備、アナログカラーによる512色中8色の選定など、バージョンアップした所が上げられますが、何と言ってもグラフィックの高速化、FM音源の使用はホビーユースにとって大きく世界を変えるものでした。

## 1-2 88シリーズの歴史

では、今日のハイスピード8ビットマシン、FH/MHが登場するまでのPC-8801シリーズを見てみましょう。



第1図 NEC PCシリーズの系列図(8ビット機のみ)



## 1 PC-8801 (56年 12月)

人気のPC-8001のソフトウェアが使用でき、しかも、640×200ドットのカラーグラフィックは大きな魅力でした。しかし、ディスクドライブがまだ高価で、周辺記憶装置としては主にカセットテープを使用。そのためにロード時間が長い、ロードエラーが起きやすいなどの欠点がありました。



## 2 PC-8801mk II (58年 11月)

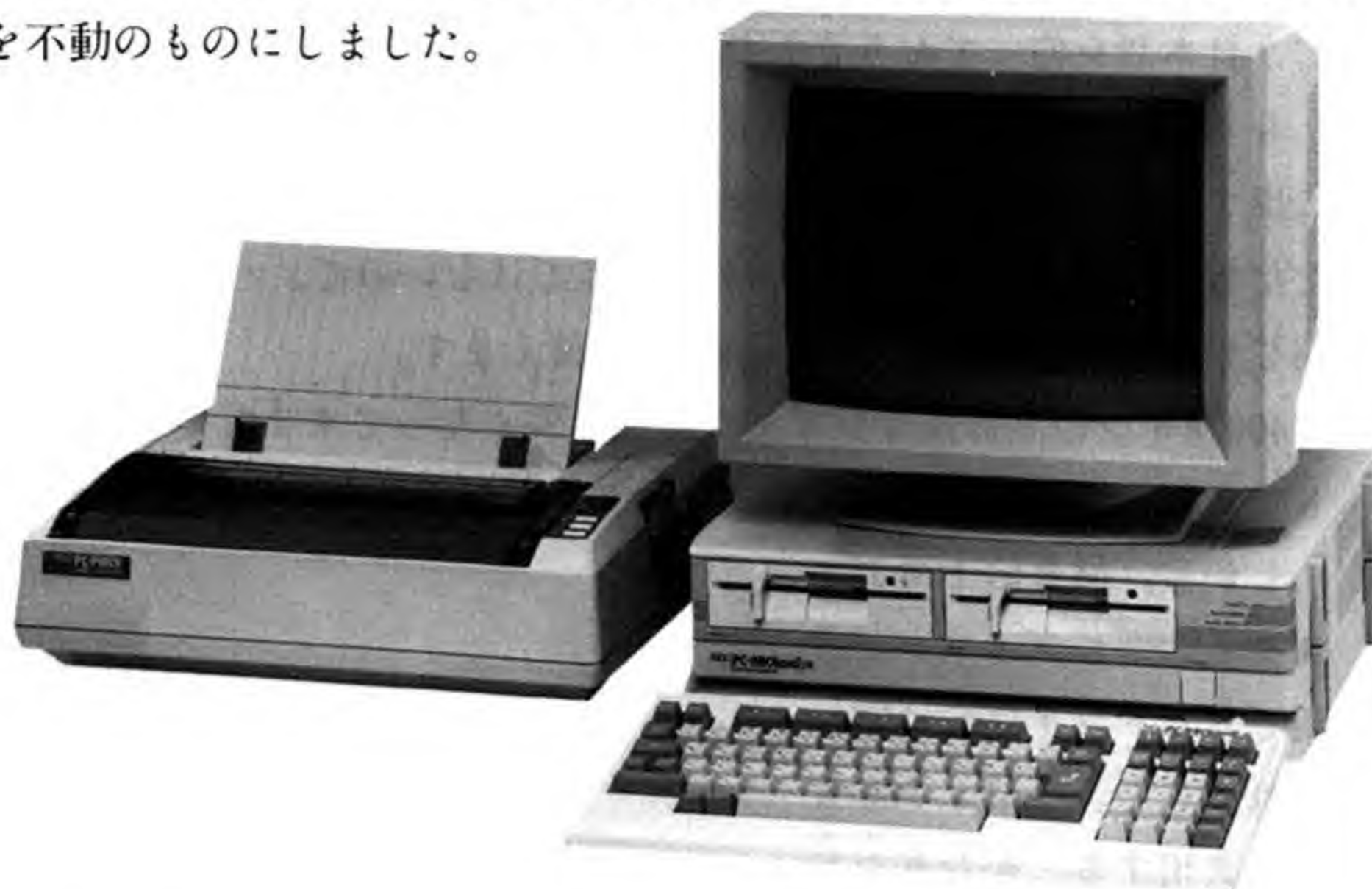
PC-8801mkIIで、コンピュータはフロッピーディスクと合体しました。パソコンの性能をフルに生かすためには、やはりフロッピーディスクは欠かせません。





### ③ PC-8801mk II SR (60年 1月)

PC-8801mkII SRが出ました。同じ時、PC-8001mkII S Rも出たのですが、こちらは320×200ドットの2画面重ね合せと言う、ゲーム屋にとってはうれしい機能までついていたのですが、ベストセラー機への壁は厚かったようです。88SRの方は順調に伸びて、88シリーズの地位を不動のものにしました。



### ④ PC-8801mk II TR (60年 9月)

PC-8801mkII TRにはなんと電話機がついてしまいました。電話回線によるコンピュータネットワーク、PC-VANは皆さんご存知でしょう。ハレー彗星接近の時にハレー情報を流していましたね（私は思わずモデムを買ってしまいました）。現在の所、パーソナルユースでのコンピュータネットワークでは、株式情報や受験情報が活躍していますが、電子メールやホームバンキングなど、情報網の可能性はまだまだこれからですね。

TRはその先がけといった機種でしょう。





## 5 PC-8801mk II FR/MR (60年 11月)

FR/MRには日本語BASICがオマケについて、漢字ROMが元気になりました。“日本語BASIC”と聞いて、命令が日本語で行われるのかと思った私です（アーハズカシイ。しかし、以前にさるメーカーのゲームコンピュータでは、“IF”のかわりに“もし”などと言う日本語を使っておりました）。

もちろん、文章中に直接漢字を入れられるようになったわけで、SRですと漢字を画面に出すのに、いちいち漢字コードが必要だったのに比べ、ずっと使いやすくなりました。ただし、日本語BASICはハードの機能ではないので、ハード的にFRとSRはほとんど同じですから、SRでもこのBASICは使えます。

400ラインの高解像度ディスプレイも安くなって来ましたし、24ピンヘッドのプリンタも安くなりましたから、簡単なワープロとしても充分実用できるようになりました。

しかし、MRの1Mディスクと320Kディスクが共用できるドライブには驚かされました。1Mと320Kとを共用する、かなりシン・ドイ事をしているわけですから、大したものですよ。ただ、MRで書き込んだ2D(320K)のディスクを、FRなどで読もうとすると読めない事があるそうで、そこまではまだ苦しい所なのではないでしょうか（ウチでやってみた所、読み込めました）。



▲PC-8801mk II FR



▲PC-8801mk II MR



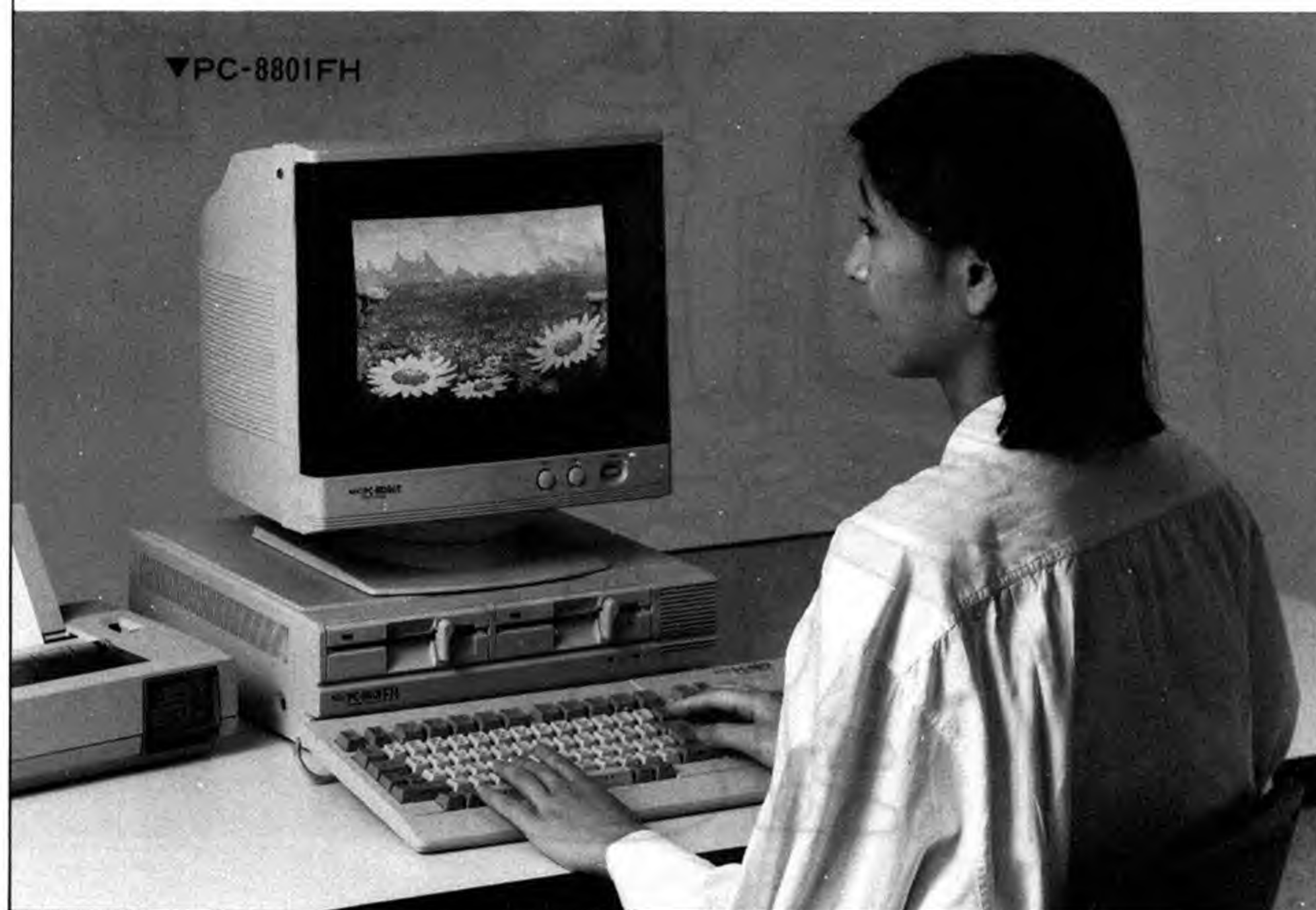
## ⑥ PC-8801FH/MH (61年 11月)

FH/MHでさらにスピードアップ、Z80の上位バージョンに8MHzで動作するものがありましたし、兄貴分の98シリーズが、8MHzから10MHzへとどんどんクロックアップして行く中で、88ももしかしたら……とのウワサがありましたが、ついに8MHzへ変身。

実質クロックとしては6MHz程度と思われそうですが、コンピュータは処理が早ければ早い程良いわけで、ここまで来れば、初期の頃の16ビットマシンと同じ位の性能では？という感を受けます。



▲PC-8801MH



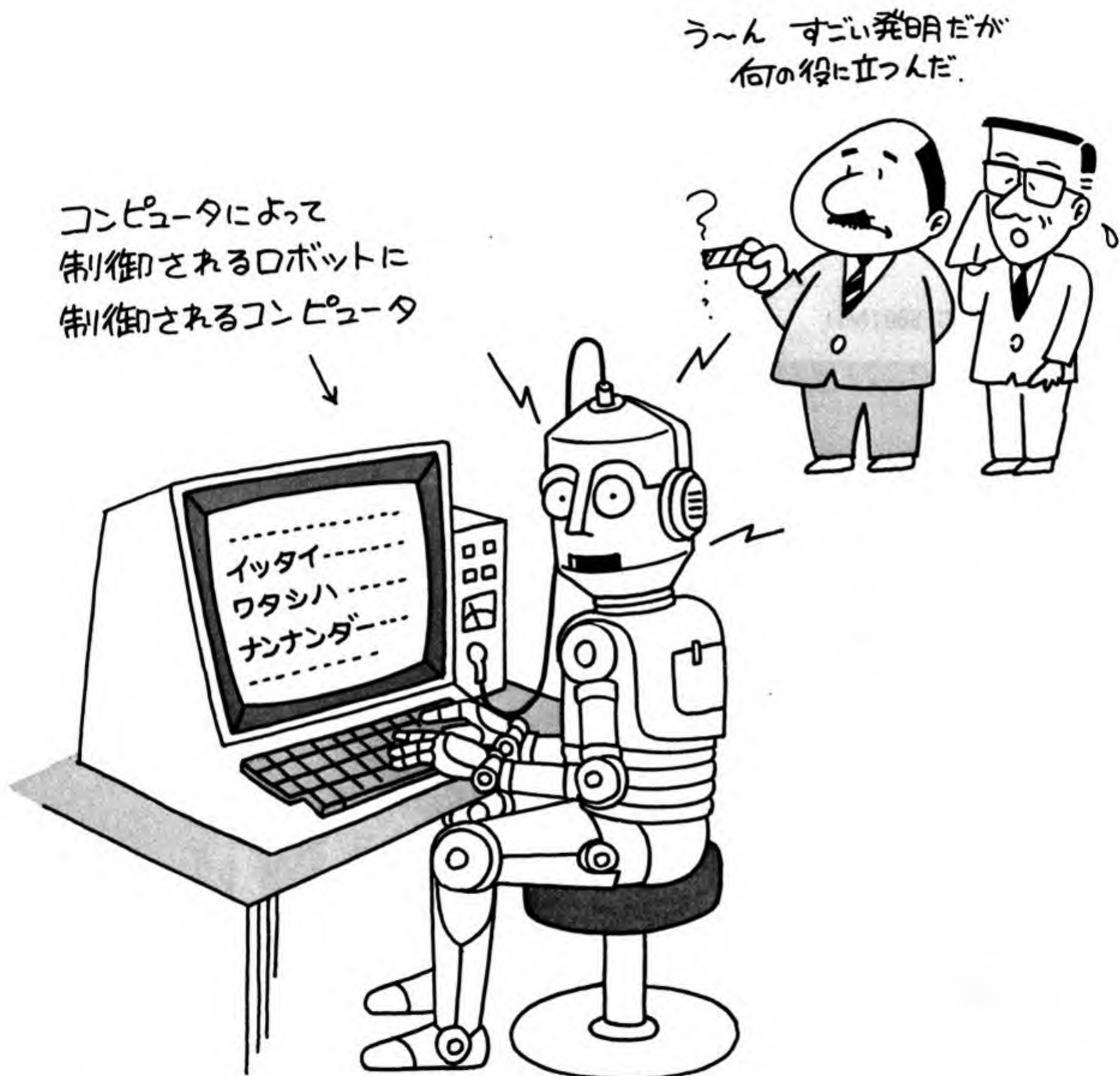
▼PC-8801FH



この様に88シリーズでも登場以来5年以上たちますが、性能アップ、価格ダウンを続けてここまで来ました。しかし、大幅な変更は、SRになった時で、ソフトウェアではSR以降共通です。

どんなパソコンでも、それを知り、使いこなせるようになるまでには、ある程度の時間が必要です。せっかく利用できるようになった技術が、次のモデルチェンジで使えなくなってしまったと言うのでは何にもなりません。しかし、今までの機種の流れでは、前の性能をほぼ受けつぎ、さらに拡張機能を持たせる(上位コンパチ)ようになっていますから、習得した技術は今後も生かせるでしょう。

第二章以降は、具体的に88を活用するためのテクニックを紹介しています。あなたの88がより身近なものになるようにがんばって下さい。





第 2 章

2

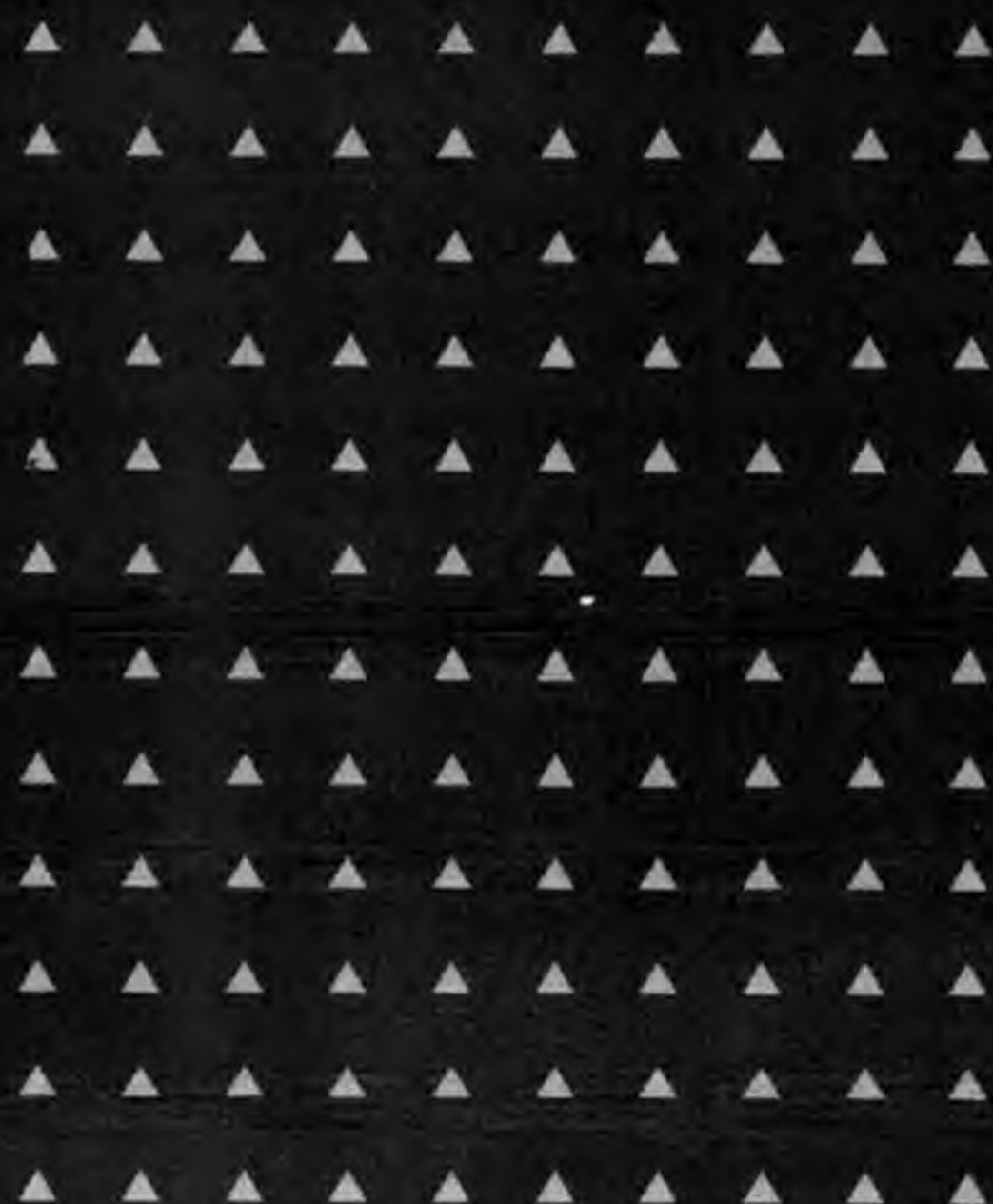
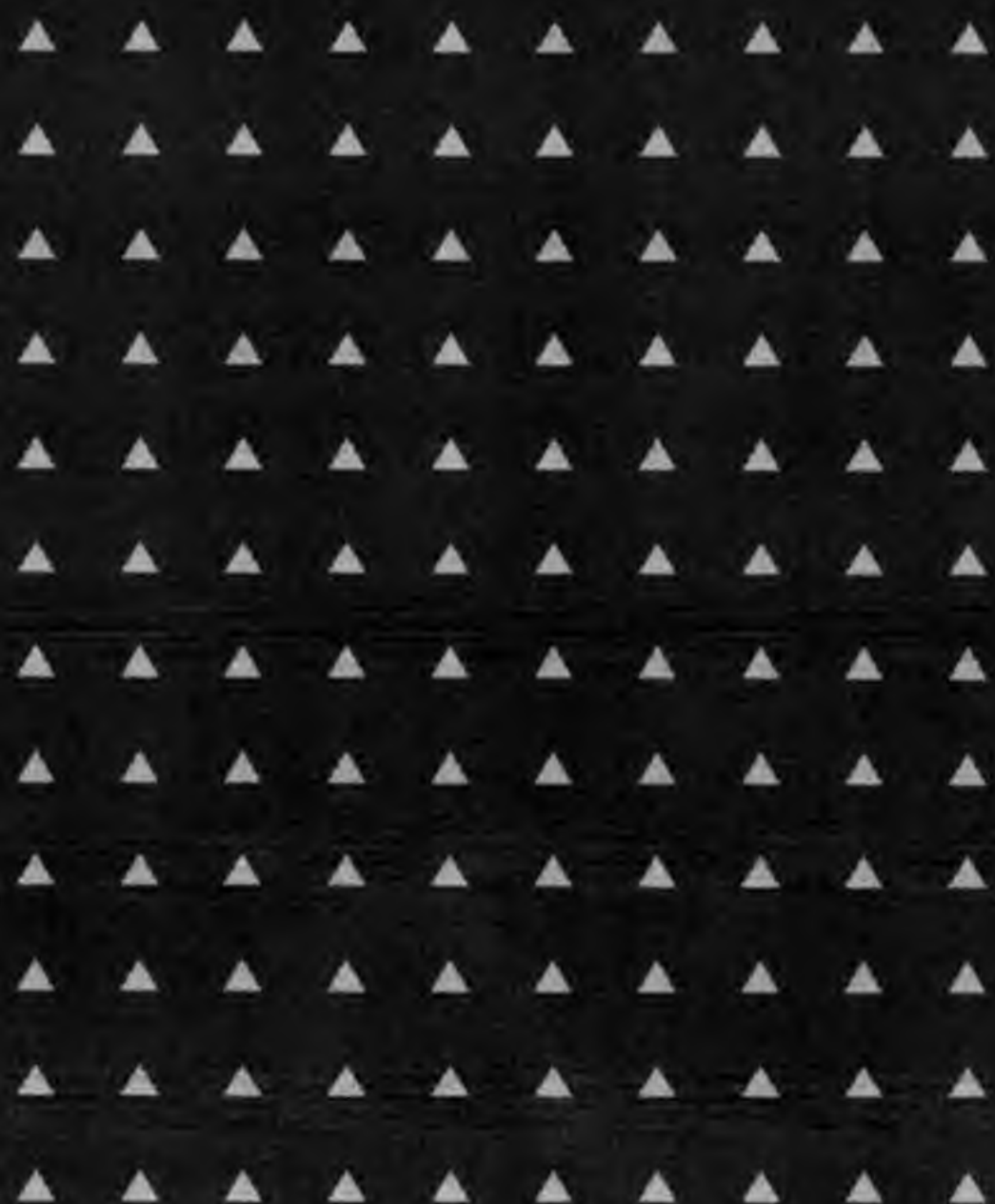
# IOの実戦的活用

2-1 キースキャン

2-2 画面コントロール

2-3 パレットコントロール

2-4 フロッピーディスクコントロール







## 2-1 キースキャン

人間がコンピュータにコンタクトする部分がキーですね。プログラムでキーをスキャンする事ができなければ、何も始まりません。

BASICでキーを受けつけるには、INPUT文、INKEY\$文などがありますが、INP関数を使ってキー入力をさせて見た事がありますか？

INP関数はこの章の目的であるI/Oのうち、ポートへの入力を調べるものです。

PC-88のキーは、ポートに直接つながっており、従ってINP関数でキー入力の様子をリアルタイムで調べる事ができるのです。

次のBASICプログラムを入力してみてください。

```
10 A=INP(0)
```

```
20 PRINT HEX$(A):GOTO 10
```

RUN☑として、このプログラムを走らせると、画面左側に“FF”が並ぶと思います。

ここで、テンキーの“0”を押してみてください。FFが、“FE”に変わると思います。カナキーと一緒にいる“0”ではこの様な事は起こりません。

“1”を押すと“FD”になりますね、他のキーもいろいろと押してみてください。テンキーの0～7は変化しますが、他のキーではFFのままになるでしょう。

他のキーをスキャンするには、INP(0)の0を他の数字に変えてやります。テンキーの8を調べるには、INP(1)として、プログラムを走らせ、8を押すと、FFがFEに変化するでしょう。

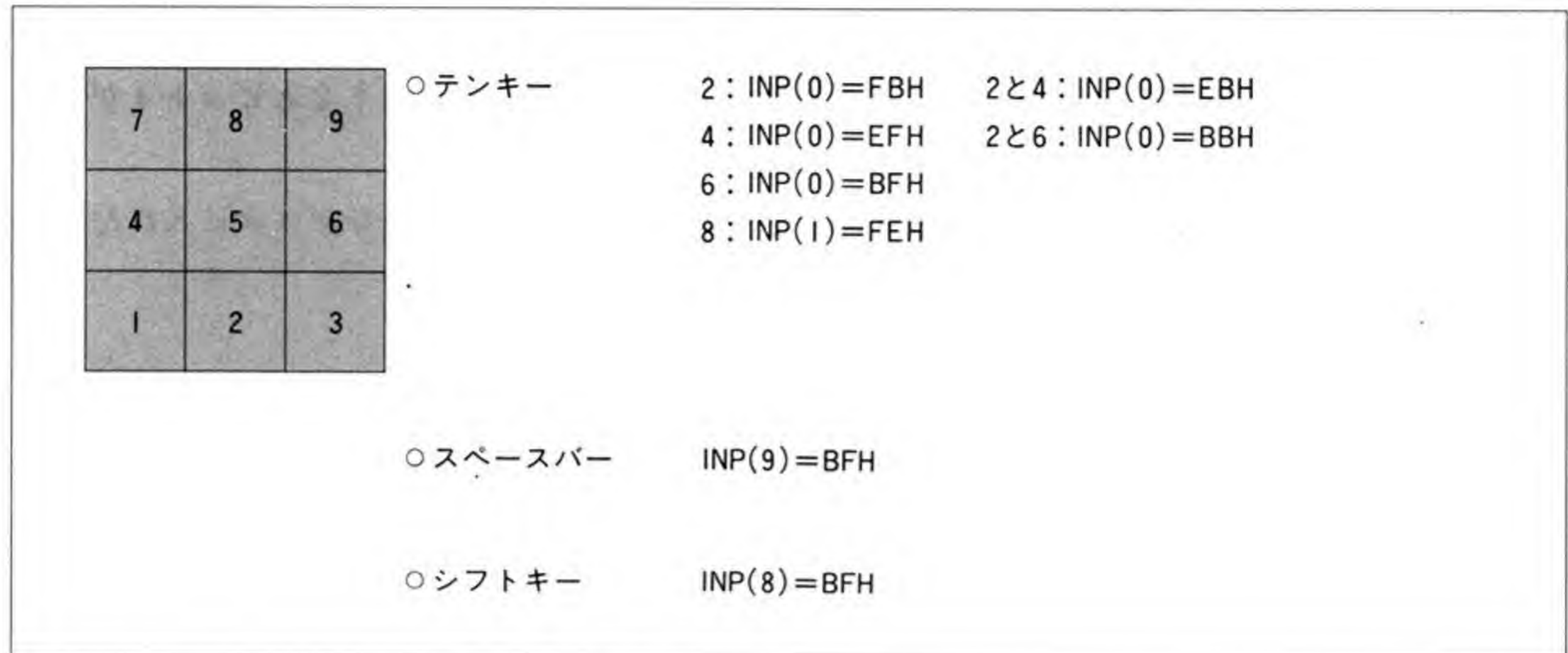
ですから、8と6が同時に押されたかどうかを知るには、INP(0)とINP(1)を見ればよいわけです。

ゲームでよく使うキーと、INP関数の値(16進数)を第2-1-1図に示しておきます。

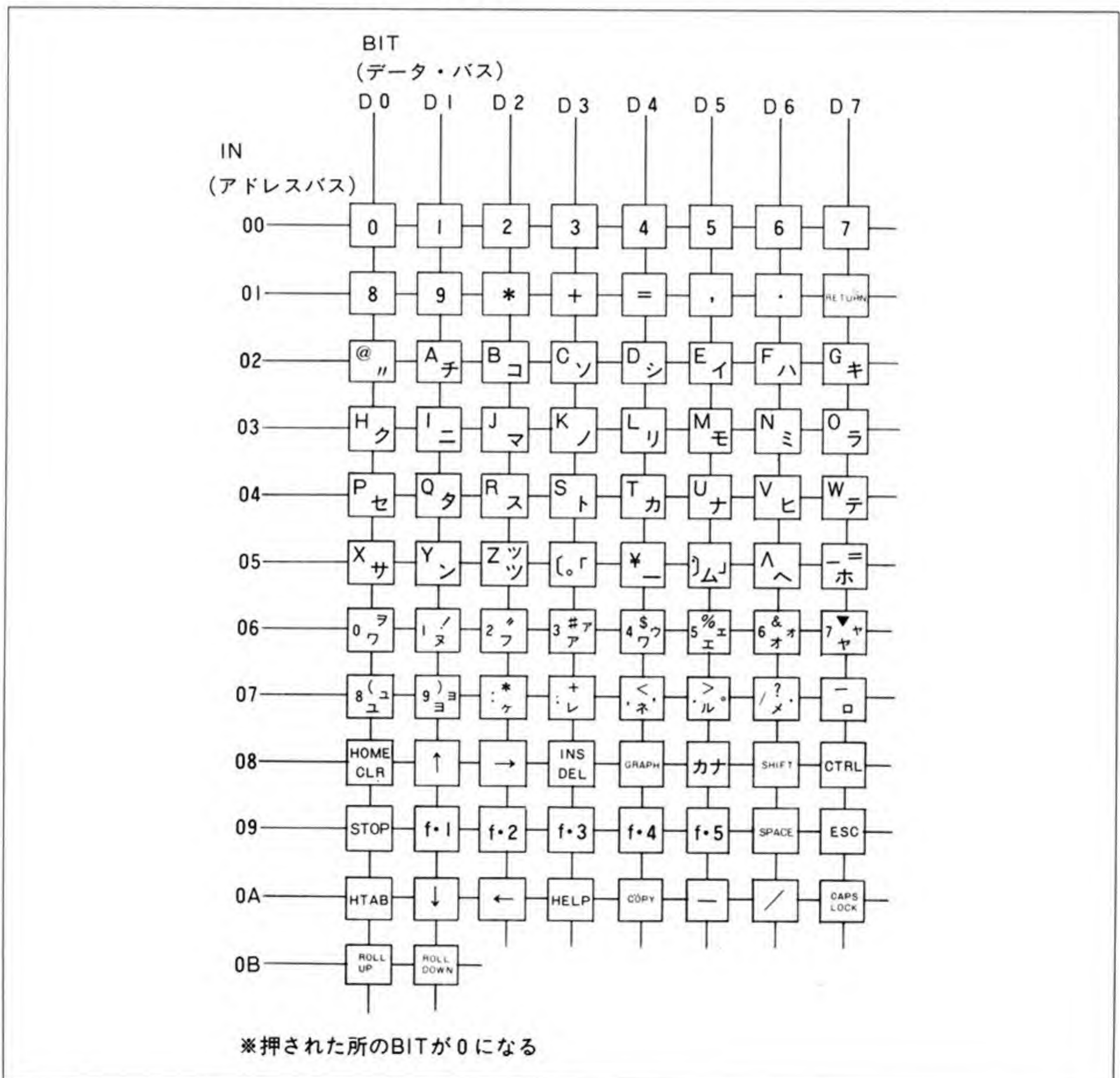
このような事が見られるのは、前にも書いた様に、キーがポートに直接つながっているからで、第2-1-2図の縦がポート番号、横がデーターのビットになっていると考えればよいでしょう。

キーは押されていないければ、そのビットは1になっています。たとえばテンキーの0が押さ





第2-1-1図 キーを押した時のポートの変化



第2-1-2図 キーマトリクス



れると、ポート0に入ってくるデータは0ビットが0で他は1になりますから、11111110の2進数で、16進ではFEHです。もし、2と4が同時に押されるなら、ビット2とビット4が0になります。11101011は16進数でEBHですね。

さて、INP関数はマシン語ではIN命令と同じでした。ポート0の値をAレジスタに入れたい時は、

ニーモニック IN A, (0)

マシンコード DB 00

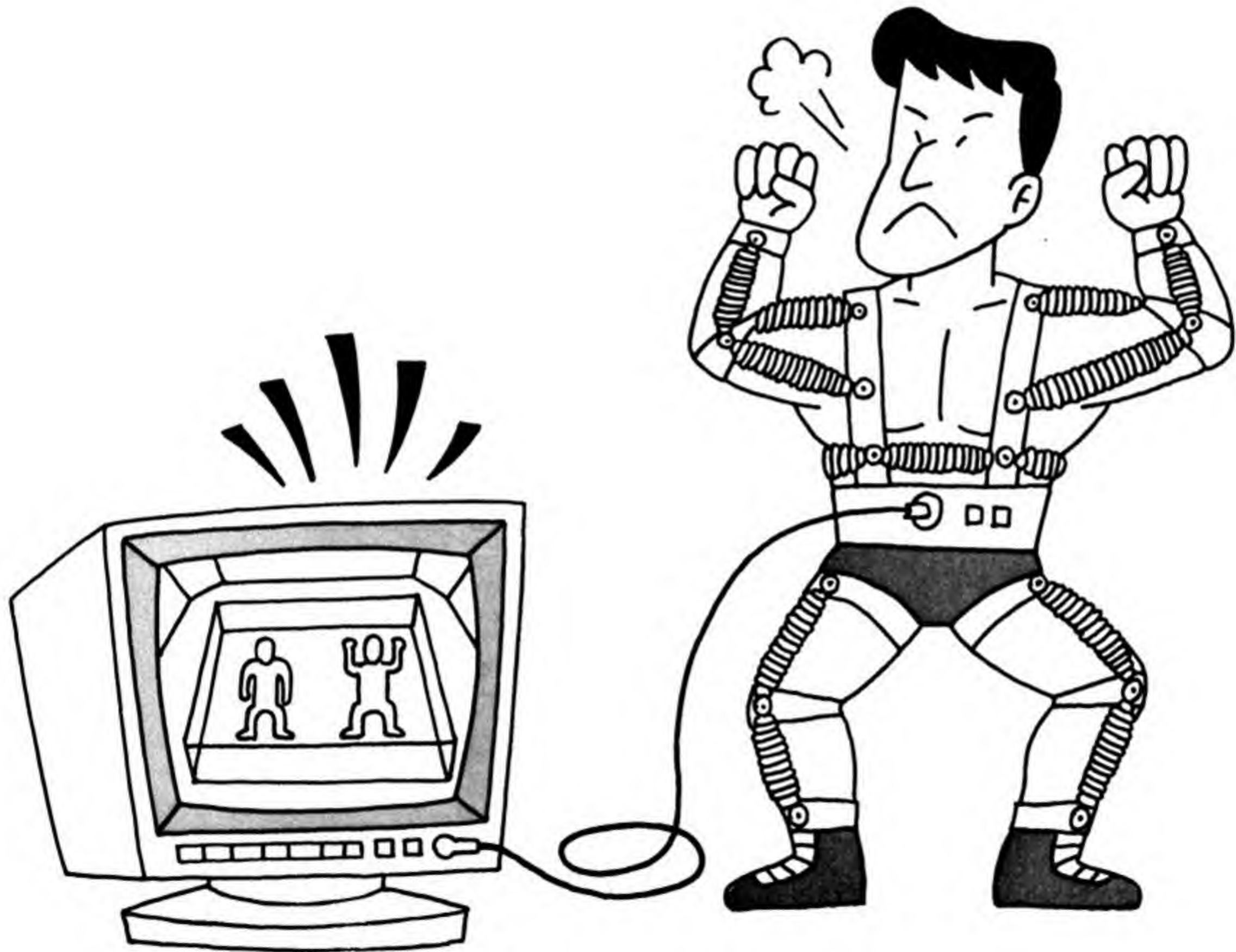
で表わせます。従って、キー“2”が押された時、キャラクタを下へ移動したい時などは、

IN A, (0)

CP 0FBH

JP Z, 下へ移動

と言う様なプログラムになります。さらに具体的なものは次の章で御紹介いたします。



闘将ダイモス式ゲーム入力装置





## 2-2 画面コントロール

画面上にグラフィックキャラクタを表示するのは、BASICのPUT@文があります。これをこれからマシン語で処理してみましょう。

グラフィックの処理はパソコンの構造により様々でPC-88でもSR以降、その処理速度は格段に速くなっています。それはハード的タイミング処理もさることながら、コントローラなどの力による所も大です。ですから、グラフィックの処理にOUT命令は欠かせません。

ここでは、グラフィックキャラクタの様々な表示方法を実戦的に行います。

### I キャラクタを作る

まず、基礎知識の確認をしておきましょう。私達が画面で見ているのは、メモリの内容そのものなのです。C000番地からFF FF番地までのメモリ空間には、実に4枚ものメモリがだぶって用意されています。一枚はメインメモリで、ここには通常BASICのシステムワークエリア、テキスト画面用のメモリ、スタック、割り込み用テーブルなどがあり、他の三枚がグラフィック用メモリです。

カラーモードでは、三枚のグラフィックメモリが青、赤、緑の色に割り当てられ、このメモリの中のビットがセットされていると(1になっていると)その所が画面に点となって表れ、三枚合成されて八つの色を表す事ができるのです。

グラフィックメモリに書いたり読んだりする方法は、他のメモリに対してする方法と何ら変わりありません。単にC000以後のメモリを四つのうちから一つ選択してやればよいのです。当然、グラフィック用のプログラムはBF FF以前に置くのが普通です。

さて、従来のPC-88や、SRのノーマルモードでは、メモリの選択(バンク切り替え)をOUT命令により行います。

OUT (5CH), A : “青”のメモリ



OUT (5DH), A : “赤”のメモリ

OUT (5EH), A : “緑”のメモリ

OUT (5FH), A : メインメモリ

Aの値は何でもかまいません。5C~5Fのポートに何かを出力する動作をすると、メモリが切り替わる様に設計されています。

ですから、BASICのPSET文等を使って画面にキャラクタを作っておき、グラフィックメモリの内容を他の場所に移してやればキャラクタデータの出来上り。また、キャラクタデータをグラフィックメモリに戻せば、キャラクタをPUTした事になります。

では、リスト2-2-1を見て下さい。縦16ライン、横32ドット(4バイト)のキャラクタを作るプログラムです。

このプログラムを走らせるには、リスト2-2-2を、BSAVE “GRGET”, &HBF00, &H100☐としてディスクにセーブしている必要があります。

プログラムを使用するには、0から7のキーでドットの色を指定します。8のキーを押すと、左右が反転します。

キャラクタを作り上げたら、“ESC”キーを押すと、キャラクタをデータに変えます。データの先頭アドレスを入力して下さい。

また、すでにデータにあるキャラクタを作り直すには、“G”を押します。

キャラクタの大きさを変えたい時は、1040行のxとyの値を変えて下さい。ただし、拡大したものが、画面からはみ出すことのないようにして下さい。

それに、“G”コマンドは16ライン32ドット専用ですので、これを使う時にはプログラムを書き替える必要があります。

スピードが遅くまだ不完全なものですので、使い易いようにみなさんで直して下さい。

#### リスト2-2-1 16ライン4バイト・キャラクタメイカー

```

1000 DEFINT A-Z:WIDTH 80,25:CONSOLE 0,25,0,1
1010 CLEAR ,&HBEFF:BLOAD "GRGET":DEFINT A-Z
1020 INIT
1030 GRGET=&HBF00
1040 X=4:Y=16:XX=X*8-1:YY=Y-1
1050 DIM C(XX,YY)
1060 SCREEN 0,0:CLS 3
1070 XC=0:YC=0
1080 INKEY
1090 LOCATE XC,YC:A=ASC(INPUT$(1))
1100 IF (A>=&H30) AND (A<=&H38) THEN 1260
1110 IF (A>=&H1C) AND (A<=&H1F) THEN 1150
1120 IF A=&H1B THEN *BLOK
1130 IF A=&H47 OR A=&H67 THEN *GEMEMO
1140 GOTO 1090
1150 IF A<>&H1D THEN 1180
1160 XC=XC-1:IF XC=-1 THEN XC=XX
1170 GOTO 1090
1180 IF A<>&H1C THEN 1210
1190 XC=XC+1:IF XC=XX+1 THEN XC=0
1200 GOTO 1090
1210 IF A<>&H1E THEN 1240
1220 YC=YC-1:IF YC=-1 THEN YC=YY
1230 GOTO 1090
1240 YC=YC+1:IF YC=YY+1 THEN YC=0
1250 GOTO 1090

```



```

1260 PSET
1270 C=A-&H30:IF C=8 THEN *HANTEN
1280 IF C=0 THEN PRINT ' '; ELSE COLOR C:PRINT '●';
1290 COLOR 7
1300 C(XC,YC)=C:PSET(400+XC,100+YC),C: POINT SET
1310 GOTO 1190
1320 REMAKE
1330 CLS 3
1340 FOR J=0 TO YY
1350 FOR I=0 TO XX
1360 LOCATE I,J:IF C(I,J)=0 THEN PRINT ' '; ELSE COLOR C(I,J):PRINT '●';:PSET(4
00+I,100+J),C(I,J)
1370 NEXT:NEXT
1380 LOCATE 0,0:XC=0:YC=0
1390 GOTO 1090
1400 *BLOK
1410 LOCATE 0,23:COLOR 7:PRINT 'DATA SEND TO MAIN RAM OK ? (Y/N)'
1420 K$=INKEY$:IF K$="" THEN 1420
1430 IF K$='Y' OR K$='y' THEN 1460
1440 LOCATE 0,23:PRINT ' '
1450 GOTO 1090
1460 LOCATE 0,23:PRINT ' '
1470 CLS 3:FOR J=0 TO YY:FOR I=0 TO XX
1480 PSET(I,J),C(I,J)
1490 NEXT:NEXT
1500 LOCATE 0,23:INPUT 'テンソク ハンチハ ? (C000-DFFF)';TEN$
1510 TEL$=RIGHT$(TEN$,2):TEH$=LEFT$(TEN$,2)
1520 POKE &HBFF0,VAL('&H'+TEL$)
1530 POKE &HBFF1,VAL('&H'+TEH$)
1540 POKE &HBFF4,VAL('&H00')
1550 POKE &HBFF5,VAL('&HC0')
1560 POKE &HBFF2,Y : 'Y SIZE
1570 POKE &HBFF3,X : 'X SIZE
1580 CALL GRGET
1590 GOTO 1330
1600 *HANTEN
1610 LOCATE 0,23:COLOR 7:PRINT 'ハンチン OK ? (Y/N)'
1620 K$=INKEY$:IF K$="" THEN 1620
1630 IF K$='Y' OR K$='y' THEN 1660
1640 LOCATE 0,23:PRINT ' '
1650 GOTO 1090
1660 LOCATE 0,23:PRINT ' '
1670 FOR JJ=0 TO YY
1680 FOR II=0 TO XX/2-1
1690 STOCK=C(II,JJ)
1700 C(II,JJ)=C(XX-II,JJ)
1710 C(XX-II,JJ)=STOCK
1720 NEXT:NEXT
1730 GOTO 1330
1740 *GEMEMO: '16 LINE 4 BYTES ONLY
1750 LOCATE 0,23:COLOR 7:PRINT 'GET MEMORY OK ? (Y/N)'
1760 K$=INKEY$:IF K$="" THEN 1760
1770 IF K$='Y' OR K$='y' THEN 1800
1780 LOCATE 0,23:PRINT ' '
1790 GOTO 1090
1800 LOCATE 0,23:INPUT 'メモリ ハンチハ ? (C000-DFFF)';MET$
1810 MET=VAL('&H'+MET$)
1820 FOR JM=0 TO 15
1830 FOR IM=0 TO 3
1840 DAMB=MET+12*JM+IM:DAMR=MET+12*JM+IM+4:DAMG=MET+12*JM+IM+8
1850 MEB=PEEK(DAMB):MER=PEEK(DAMR):MEG=PEEK(DAMG)
1860 FOR KM=0 TO 7
1870 MEB1=SGN(MEB AND 2^(7-KM)):MER1=SGN(MER AND 2^(7-KM)):MEG1=SGN(MEG AND 2^(
7-KM))
1880 C(IM*8+KM,JM)=MEB1+MER1*2+MEG1*4
1890 NEXT:NEXT:NEXT
1900 GOTO 1330

```



## リスト2-2-2a キャラクタメイカー・アセンブルリスト

```

;
; CHARACTER GET
;
;
;
;
;
;
; ORG 0BF00H
;
BF00 F3      CGET: DI          ;ワリコミ タメ
BF01 2AF0BF   LD HL,(0BFF0H)   ;READ CHR DATA ADDRESS
BF04 ED5BF4BF LD DE,(0BFF4H)   ;READ G RAM ADDRESS
BF08 ED4BF2BF LD BC,(0BFF2H)   ;READ X=B Y=C
;
BF0C D5      CGETL: PUSH DE    ;LOOP START
BF0D C5      PUSH BC
BF0E D35C    CGETB: OUT (5CH),A ;SELECT BLUE RAM
BF10 1A      LD A,(DE)         ;READ CHR DATA
BF11 D35F    OUT (5FH),A       ;SELECT MAIN RAM
BF13 77      LD (HL),A         ;SET CHR DATA
BF14 23      INC HL            ;CHR ADDRESS+1
BF15 13      INC DE            ;G ADDRESS+1
BF16 10F6    DJNZ CGETB        ;X カイ クリカエ
BF18 C1      POP BC            ;X,Y SIZE モトメ
BF19 D1      POP DE            ;G サマヨウ モトメ
;
BF1A D5      PUSH DE           ;オナシ コト ノ クリカエ
BF1B C5      PUSH BC
BF1C D35D    CGETR: OUT (5DH),A ;SELECT RED RAM
BF1E 1A      LD A,(DE)         ;
BF1F D35F    OUT (5FH),A       ;SELECT MAIN RAM
BF21 77      LD (HL),A
BF22 23      INC HL
BF23 13      INC DE
BF24 10F6    DJNZ CGETR
BF26 C1      POP BC
BF27 D1      POP DE
;
BF28 D5      PUSH DE           ;オナシ コト ノ クリカエ
BF29 C5      PUSH BC
BF2A D35E    CGETG: OUT (5EH),A ;SELECT GREEN RAM
BF2C 1A      LD A,(DE)         ;
BF2D D35F    OUT (5FH),A       ;SELECT MAIN RAM
BF2F 77      LD (HL),A
BF30 23      INC HL
BF31 13      INC DE
BF32 10F6    DJNZ CGETG
BF34 C1      POP BC
BF35 D1      POP DE
;
BF36 E5      PUSH HL
BF37 215000   LD HL,50H
BF3A 19      ADD HL,DE          ;NEXT LINE
BF3B EB      EX DE,HL
BF3C E1      POP HL
BF3D 0D      DEC C
BF3E 20CC    JR NZ,CGETL
BF40 FB      EI                ;ワリコミ OK !
BF41 C9      RET
;

```

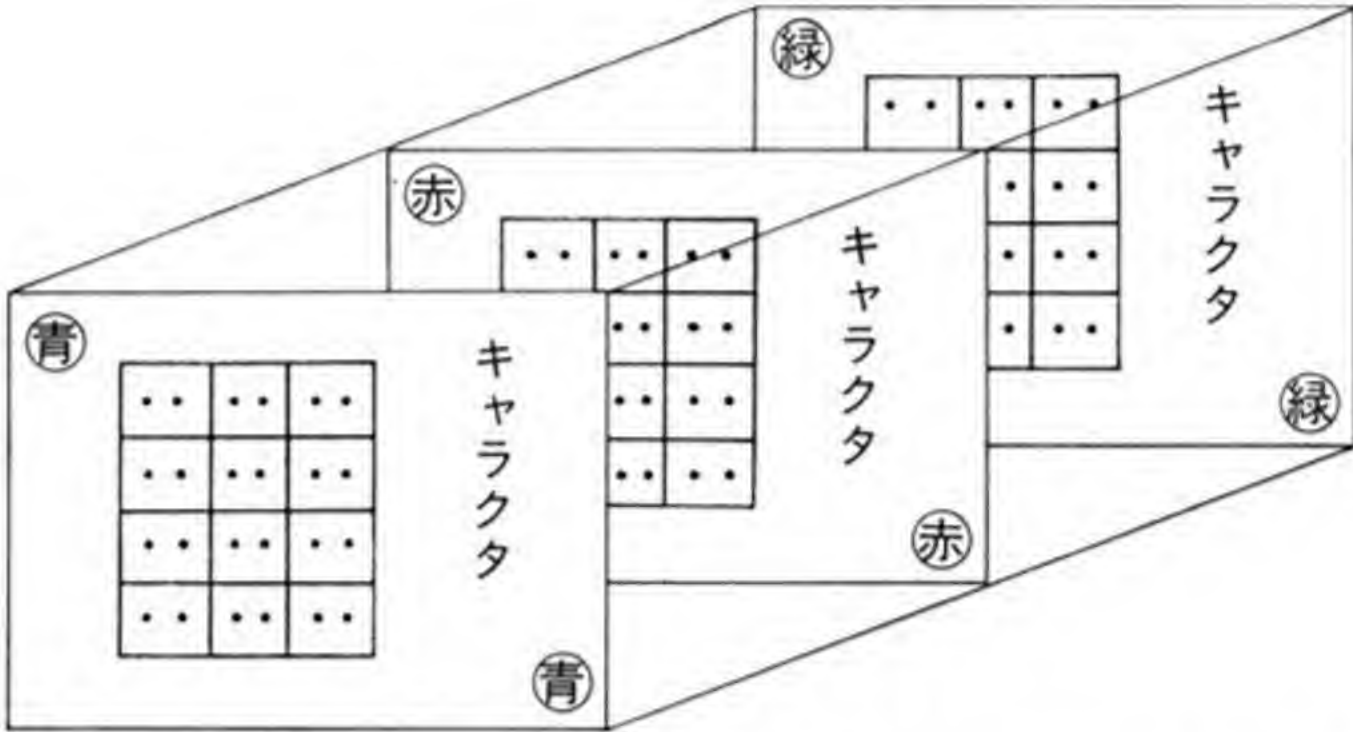


リスト2-2-2b 入力用プログラム

BF00	F3	2A	F0	BF	ED	5B	F4	BF	ED	4B	F2	BF	D5	C5	D3	5C
BF10	1A	D3	5F	77	23	13	10	F6	C1	D1	D5	C5	D3	5D	1A	D3
BF20	5F	77	23	13	10	F6	C1	D1	D5	C5	D3	5E	1A	D3	5F	77
BF30	23	13	10	F6	C1	D1	E5	21	50	00	19	EB	E1	0D	20	CC
BF40	FB	C9	00	00	00	00	00	00	00	00	00	00	00	00	00	00

II キャラクタをデータにする

さて、重要なのはリスト2-2-2の方です。このプログラムは画面にあるキャラクタをデータ列に変換するもので、リスト2-2-1あるいは他の何かを使って画面に書いた絵を、青、赤、緑各1列ずつ縦のライン数分並べていきます(第2-2-1図)。



グラフィックは、青、赤、緑のメモリにあるビットパターンでできている。これを16進数でのぞくと下の様にみえます。

青

56H	11H	10H
.....	.....	.....
.....	.....	.....

赤

11H	20H	E0H
.....	.....	.....
.....	.....	.....

緑

54H	18H	C0H
.....	.....	.....
.....	.....	.....

リスト2-2-2のプログラムはこの三つのメモリのデータをメモリ上に次の様に並べます。

56H	11H	10H	11H	20H	E0H	54H	18H	C0H			
青一列			赤一列			緑一列			次の青一列		次の赤一列

上の列では横3バイトずつですが、リスト2-2-1のキャラクタメイカーでは横4バイトずつ16ライン分のメモリを一列に並べますから

$4 \times 3 \times 16 = 192$  (C0H)

1キャラクタにつき192バイトのデータ量となります。

第2-2-1図 グラフィックデータの構造

23







```

BFD5 D1          POP  DE
;
BFD6 E5          PUSH HL
BFD7 215000      LD   HL,50H
BFDA 19          ADD  HL,DE          ;NEXT LINE
BFDB EB          EX   DE,HL
BFDC E1          POP  HL
BFDD 0D          DEC  C
BFDE 20CC        JR   NZ,LPUTL
BFE0 FB          EI
BFE1 C9          RET
;
;

```

## リスト2-2-3b テスト用キャラクタ表示プログラムとサンプルキャラクタデータ

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
BF00:	21	00	C0	22	F0	BF	22	F4	BF	06	04	C5	E5	06	08	C5	:0E
BF10:	CD	A0	BF	2A	F0	BF	11	C0	00	19	22	F0	BF	2A	F4	BF	:9D
BF20:	11	06	00	19	22	F4	BF	C1	10	E5	E1	11	00	0A	19	22	:F2
BF30:	F4	BF	C1	10	D6	C9	00	00	00	00	00	00	00	00	00	00	:23
BF40:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BF50:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BF60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BF70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BF80:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BF90:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BFA0:	F3	2A	F0	BF	ED	5B	F4	BF	ED	4B	F2	BF	D5	C5	7E	D3	:9B
BF80:	5C	12	D3	5F	23	13	10	F6	C1	D1	D5	C5	7E	D3	5D	12	:C8
BFC0:	D3	5F	23	13	10	F6	C1	D1	D5	C5	7E	D3	5E	12	D3	5F	:8D
BFD0:	23	13	10	F6	C1	D1	E5	21	50	00	19	EB	E1	0D	20	CC	:02
BFE0:	FB	C9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:C4
BFF0:	00	C0	10	04	00	C0	00	00	00	00	00	00	00	00	00	00	:94
sum :	33	9C	46	A0	B9	30	9C	1C	A2	E5	65	08	36	F1	E3	B6	:0A

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C000:	00	0F	FE	00	00	03	F8	00	00	0F	FE	00	00	7F	FF	80	:13
C010:	00	0C	00	00	00	7F	FF	80	01	FE	00	60	00	70	00	00	:D9
C020:	01	FE	00	60	03	FC	00	10	00	E0	7E	10	03	FC	7E	10	:69
C030:	07	FC	00	08	01	E0	E6	08	07	FC	E6	08	07	FC	00	08	:D6
C040:	01	E0	E7	08	07	FC	E7	08	07	FC	00	08	01	E0	FF	C0	:6D
C050:	07	FC	FF	C8	00	00	00	10	00	00	FF	00	05	54	FF	10	:41
C060:	01	E7	3F	E0	1C	41	01	E0	1D	E7	3F	E0	01	E7	07	E0	:37
C070:	1C	41	01	E0	1D	E7	07	E0	01	E7	83	E0	1C	41	F1	E3	:A5
C080:	1D	E7	83	E0	01	E7	83	E0	1C	40	71	E3	1D	E7	83	E0	:C9
C090:	07	F0	07	FC	00	40	01	FF	07	F0	07	FC	3F	FF	FF	F8	:69
COA0:	FF	C0	00	06	3F	FF	FF	F8	3F	F0	00	00	C0	00	00	00	:E9
COB0:	3F	F0	00	00	00	00	00	00	7C	00	00	00	00	00	00	00	:AB
COC0:	00	7F	F0	00	00	1F	C0	00	00	7F	F0	00	01	FF	FE	00	:BB
COD0:	00	00	30	00	01	FF	FE	00	06	00	7F	80	00	00	0E	00	:41
COE0:	06	00	7F	80	08	00	3F	C0	08	7E	07	00	08	7E	3F	C0	:1E
COF0:	10	00	3F	E0	10	67	07	80	10	67	3F	E0	10	00	3F	E0	:F2
sum :	A5	1F	8C	3A	9D	2D	53	87	29	37	50	7F	62	A6	7F	A3	:87

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C100:	10	E7	07	80	10	E7	3F	E0	10	00	3F	E0	03	FF	07	80	:4C
C110:	13	FF	3F	E0	08	00	00	00	00	FF	00	00	08	FF	2A	A0	:09
C120:	07	FC	E7	80	07	80	82	38	07	FC	E7	B8	07	E0	E7	80	:9B
C130:	07	80	82	38	07	E0	E7	B8	07	C1	E7	80	C7	8F	82	38	:06
C140:	07	C1	E7	B8	07	C1	E7	80	C7	8E	02	38	07	C1	E7	B8	:8C



C150:	3F	E0	0F	E0	FF	80	02	00	3F	E0	0F	E0	1F	FF	FF	FC	:B6
C160:	60	00	03	FF	1F	FF	FF	FC	00	00	0F	FC	00	00	00	03	:89
C170:	00	00	0F	FC	00	00	00	00	00	00	00	3E	00	00	00	00	:49
C180:	00	0F	FE	00	00	03	F8	00	00	0F	FE	00	00	7F	FF	80	:13
C190:	00	16	00	00	00	7F	FF	80	01	FE	00	60	00	48	00	00	:BB
C1A0:	01	FE	00	60	03	FC	00	10	01	90	7E	10	03	FC	7E	10	:1A
C1B0:	07	FC	00	08	03	20	E6	08	07	FC	E6	08	07	FC	00	08	:18
C1C0:	03	20	E7	08	07	FC	E7	08	07	FC	00	08	03	20	FF	C0	:F1
C1D0:	07	FC	FF	C8	00	00	00	10	00	00	FF	00	05	54	FF	10	:41
C1E0:	01	F8	FF	E0	1D	20	06	60	1D	F8	FF	E0	01	E3	CF	E0	:02
C1F0:	1D	20	06	60	1D	E3	CF	E0	01	EF	1F	E0	1D	20	06	60	:E4
sum :	07	56	A0	23	92	24	29	3C	52	A6	AC	AA	2F	63	D0	37	:22
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C200:	1D	EF	1F	E0	01	E7	0F	E0	1D	20	C6	60	1D	E7	0F	E0	:38
C210:	07	F2	0F	E0	00	21	C6	60	07	F2	0F	E0	00	F8	1F	C0	:EE
C220:	00	00	04	C0	00	F8	1F	C0	00	3C	FF	00	00	C0	09	00	:9F
C230:	00	3C	FF	00	00	00	00	00	00	7E	3F	00	00	00	00	00	:F8
C240:	00	7F	F0	00	00	1F	C0	00	00	7F	F0	00	01	FF	FE	00	:BB
C250:	00	00	68	00	01	FF	FE	00	06	00	7F	80	00	00	12	00	:7D
C260:	06	00	7F	80	08	00	3F	C0	08	7E	09	80	08	7E	3F	C0	:A0
C270:	10	00	3F	E0	10	67	04	C0	10	67	3F	E0	10	00	3F	E0	:2F
C280:	10	E7	04	C0	10	E7	3F	E0	10	00	3F	E0	03	FF	04	C0	:C6
C290:	13	FF	3F	E0	08	00	00	00	00	FF	00	00	08	FF	2A	A0	:09
C2A0:	07	FF	1F	80	06	60	04	B8	07	FF	1F	B8	07	F3	C7	80	:E5
C2B0:	06	60	04	B8	07	F3	C7	B8	07	F8	F7	80	06	60	04	B8	:33
C2C0:	07	F8	F7	B8	07	F0	E7	80	06	63	04	B8	07	F0	E7	B8	:C7
C2D0:	07	F0	4F	E0	06	63	84	00	07	F0	4F	E0	03	F8	1F	00	:53
C2E0:	03	20	00	00	03	F8	1F	00	00	FF	3C	00	00	90	03	00	:0B
C2F0:	00	FF	3C	00	00	00	00	00	00	FC	7E	00	00	00	00	00	:B5
sum :	7B	E8	2F	50	4F	0A	89	50	6D	74	2C	D0	58	E5	C7	90	:85
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C300:	00	0F	F0	00	00	03	F0	00	00	0F	F0	00	20	7F	FE	04	:92
C310:	20	1F	FE	04	20	7F	FE	04	11	80	01	88	10	00	01	88	:95
C320:	11	80	01	88	0A	00	00	50	08	7F	FE	50	0A	7F	FE	50	:20
C330:	06	00	00	60	04	67	E6	20	06	67	E6	60	06	00	00	60	:F0
C340:	04	E5	A7	20	06	E5	A7	60	06	00	00	60	04	FD	BF	20	:E8
C350:	06	FD	BF	60	00	00	00	00	00	FC	3F	00	06	FC	3F	60	:FE
C360:	1F	FF	FF	F8	01	3F	FF	00	1F	FF	FF	F8	3F	E1	87	FC	:0C
C370:	01	21	87	00	3F	FF	FF	FC	61	E1	87	FE	1F	21	87	00	:70
C380:	61	FF	FF	FE	21	FF	FF	86	1F	FF	FF	78	21	FF	FF	86	:3C
C390:	07	FF	FF	84	00	00	00	78	07	FF	FF	84	07	F0	0F	C0	:50
C3A0:	00	00	00	00	07	F0	0F	C0	0F	E0	00	00	00	00	07	E0	:9C
C3B0:	0F	E0	00	00	00	00	00	00	0F	E0	00	00	00	00	00	00	:DE
C3C0:	00	0F	F0	00	00	03	F0	00	00	0F	F0	00	20	7F	FE	04	:92
C3D0:	20	1F	FE	04	20	7F	FE	04	11	80	01	88	10	00	01	88	:95
C3E0:	11	80	01	88	0A	00	00	50	08	7F	FE	50	0A	7F	FE	50	:20
C3F0:	06	00	00	60	04	67	E6	20	06	67	E6	60	06	00	00	60	:F0
sum :	0F	3C	C8	D2	CA	E4	5B	02	08	84	6D	C2	10	E6	1B	1A	:D6
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C400:	04	E5	A7	20	06	E5	A7	60	06	00	00	60	04	FD	BF	20	:E8
C410:	06	FD	BF	60	00	00	00	00	00	FC	3F	00	06	FC	3F	60	:FE
C420:	1F	FF	FF	F8	01	3F	FF	80	1F	FF	FF	F8	3F	E1	87	FC	:8C
C430:	01	21	87	80	3F	FF	FF	FC	7F	E1	87	86	01	21	87	F8	:70
C440:	7F	FF	FF	86	61	FF	FF	84	1F	FF	FF	F8	61	FF	FF	84	:DE
C450:	21	FF	FF	E0	1E	00	00	00	21	FF	FF	E0	03	F0	0F	E0	:FE
C460:	00	00	00	00	03	F0	0F	E0	00	00	07	F0	07	E0	00	00	:C0
C470:	00	00	07	F0	00	00	00	00	00	00	07	F0	00	00	00	00	:EE
C480:	00	0F	F0	00	00	03	F0	00	00	0F	F0	00	20	7F	FE	04	:92



```

C490: 20 10 9E 04 20 7F FE 04 11 FF FF 88 10 40 23 88 :05
C4A0: 11 FF FF 88 0B FF FF D0 08 80 11 D0 0B FF FF D0 :B2
C4B0: 07 FF FF E0 05 00 08 E0 07 FF FF E0 07 FF FF E0 :9C
C4C0: 05 00 08 E0 07 FF FF E0 07 FF FF E0 05 00 08 E0 :A4
C4D0: 07 FF FF E0 00 00 00 00 00 00 00 00 05 55 AA A0 :89
C4E0: 1E 20 82 78 1C F3 CF 38 1E F3 CF 78 36 20 82 6C :EA
C4F0: 24 F3 CF 24 36 F3 CF 6C 76 20 82 6E 44 F3 CF 22 :1C
-----
sum : 50 2F D5 16 51 78 45 78 9F 79 20 94 7B EF 3C 22 :84

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: 76 F3 CF 6E 76 20 82 6C 44 F3 CF 20 76 F3 CF 6C :F4
C510: 37 FF FF E0 04 00 00 20 37 FF FF E0 03 F0 0F E0 :30
C520: 00 00 00 20 03 F0 0F E0 00 00 00 00 07 F0 0F F0 :F8
C530: 00 00 00 00 00 00 00 00 00 00 0F F8 00 00 00 00 :07
C540: 00 0F F0 00 00 0F F0 00 00 0F F0 00 20 7F FE 04 :9E
C550: 20 50 26 04 20 7F FE 04 11 FF FF 88 11 40 09 88 :B4
C560: 11 FF FF 88 0B FF FF D0 0A 80 04 D0 0B FF FF D0 :A7
C570: 07 FF FF E0 05 00 04 60 07 FF FF E0 07 FF FF E0 :18
C580: 05 00 04 60 07 FF FF E0 07 FF FF E0 05 00 04 60 :9C
C590: 07 FF FF E0 00 00 00 00 00 00 00 00 05 55 AA A0 :89
C5A0: 1E 20 82 78 1C F3 CF 38 1E F3 CF 78 36 20 82 6C :EA
C5B0: 24 F3 CF 24 36 F3 CF 6C 76 20 82 6E 44 F3 CF 22 :1C
C5C0: 76 F3 CF 6E 36 20 82 6E 04 F3 CF 22 36 F3 CF 6E :3A
C5D0: 07 FF FF EC 04 00 00 20 07 FF FF EC 07 F0 0F C0 :CC
C5E0: 04 00 00 00 07 F0 0F C0 00 00 00 00 0F F0 0F E0 :B8
C5F0: 00 00 00 00 00 00 00 00 1F F0 00 00 00 00 00 00 :0F
-----
sum : B4 53 04 10 47 92 B0 72 62 73 ED 04 93 CB DE 14 :2C

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: 00 00 00 00 00 07 80 00 00 07 80 00 00 00 00 00 :0E
C610: 00 0C 00 00 00 0C 07 80 00 00 00 00 00 0F 00 00 :AE
C620: 00 0F 0F 00 00 00 00 00 00 0C 00 C0 00 0C 1F 00 :15
C630: 00 00 00 00 00 0C 00 30 03 0C 3F C0 00 00 00 00 :4A
C640: 00 0C 00 18 0F 8C 7C 60 00 00 00 00 00 07 01 8C :2F
C650: 1F C7 7C 70 00 00 00 00 00 03 C0 06 3F FF DF E0 :98
C660: 00 01 C0 00 00 06 30 03 7F FF F6 00 00 00 70 00 :DE
C670: 00 0F 8E 03 78 7F FE 00 00 00 04 00 00 30 00 00 :C9
C680: E0 F0 04 00 00 00 00 00 00 30 00 00 C0 F7 F0 00 :AB
C690: 00 00 00 00 00 0F 00 00 80 7F 0E 00 00 00 00 00 :1C
C6A0: 00 00 00 00 00 0F 38 00 00 00 7F 00 00 00 7F 00 :26
C6B0: 01 E0 00 00 03 FC 00 00 03 FC 00 00 00 00 00 00 :DF
C6C0: 00 00 00 00 00 01 E0 00 00 01 E0 00 00 00 00 00 :C2
C6D0: 00 00 30 00 01 E0 30 00 00 00 00 00 00 00 F0 00 :31
C6E0: 00 F0 F0 00 00 00 00 00 03 00 30 00 00 F8 30 00 :3B
C6F0: 00 00 00 00 0C 00 30 00 03 FC 30 C0 00 00 00 00 :2B
-----
sum : 00 BE FD 8B 97 0C A9 13 0B C9 46 46 FF 40 FE 6C :AE

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: 18 00 30 00 06 3E 31 F0 00 00 00 00 31 80 E0 00 :3E
C710: 0E 3E E3 F8 00 00 00 00 60 03 C0 00 07 FB FF FC :47
C720: 00 03 80 00 C0 0C 60 00 00 6F FF FE 00 0E 00 00 :29
C730: C0 71 F0 00 00 7F FE 1E 00 20 00 00 00 00 0C 00 :E8
C740: 00 20 0F 07 00 00 00 00 00 00 0C 00 00 0F EF 03 :43
C750: 00 00 00 00 00 00 0F 00 00 70 FE 01 00 00 00 00 :5F
C760: 00 00 00 00 00 1C 0F 00 00 FE 00 00 00 FE 00 00 :27
C770: 00 00 07 80 00 00 3F C0 00 00 3F C0 00 00 00 00 :85
C780: 00 00 00 00 00 30 00 00 00 30 00 00 00 00 00 00 :60
C790: 00 33 00 00 00 33 07 80 00 00 00 00 00 1E 00 00 :0B
C7A0: 00 1E 0F 00 00 00 00 00 00 0C 00 C0 01 8C 1F 00 :A5
C7B0: 00 00 00 00 00 0C 00 30 07 8C 3F C0 00 00 00 00 :CE
C7C0: 00 0C 00 18 0F CC 7C 60 00 00 00 00 00 07 01 8C :6F
C7D0: 1F E7 7C 70 00 00 00 00 00 03 C0 06 3F FF DF E0 :B8

```



C7E0:	00	01	C0	00	00	06	30	03	7C	FF	F6	00	00	00	70	00	:DB
C7F0:	00	0F	8E	03	70	7F	FE	00	00	00	1C	00	00	30	00	00	:D9
sum :	05	26	72	0A	45	A5	7E	E1	E3	CA	19	45	78	76	49	6B	:9D
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C800:	C0	F0	1C	00	00	00	00	00	00	30	00	00	80	F7	80	00	:F3
C810:	00	00	00	00	00	00	00	00	00	70	E0	00	00	00	00	00	:50
C820:	00	00	00	00	00	07	90	00	00	C0	03	00	00	C0	03	00	:1D
C830:	00	3C	18	00	00	7E	1E	00	00	7E	1E	00	00	00	00	00	:8C
C840:	00	00	00	00	00	00	0C	00	00	00	0C	00	00	00	00	00	:18
C850:	00	00	CC	00	01	E0	CC	00	00	00	00	00	00	00	78	00	:F1
C860:	00	F0	78	00	00	00	00	00	03	00	30	00	00	F8	31	80	:44
C870:	00	00	00	00	0C	00	30	00	03	FC	31	E0	00	00	00	00	:4C
C880:	18	00	30	00	06	3E	33	F0	00	00	00	00	31	80	E0	00	:40
C890:	0E	3E	E7	F8	00	00	00	00	60	03	C0	00	07	FB	FF	FC	:4B
C8A0:	00	03	80	00	C0	0C	60	00	00	6F	FF	3E	00	0E	00	00	:69
C8B0:	C0	71	F0	00	00	7F	FE	0E	00	38	00	00	00	00	0C	00	:F0
C8C0:	00	38	0F	03	00	00	00	00	00	00	0C	00	00	01	EF	01	:47
C8D0:	00	00	00	00	00	00	00	00	00	07	0E	00	00	00	00	00	:15
C8E0:	00	00	00	00	00	09	E0	00	00	C0	03	00	00	C0	03	00	:6F
C8F0:	00	18	3C	00	00	78	7E	00	00	78	7E	00	00	00	00	00	:40
sum :	A6	1E	4A	FB	D3	AF	A5	FE	66	C3	C8	1E	B8	F9	09	7D	:74
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C900:	00	00	00	00	08	00	00	13	08	00	00	13	00	20	04	00	:5A
C910:	0D	00	00	1C	0D	E0	07	1C	00	10	08	00	06	00	00	10	:67
C920:	06	70	0E	10	00	00	00	00	04	01	80	10	04	3E	7C	10	:F7
C930:	00	00	00	00	08	01	80	10	08	7E	7E	50	00	00	00	00	:ED
C940:	18	01	80	20	19	62	46	20	00	00	00	00	06	0D	B0	40	:9D
C950:	06	62	46	50	00	00	00	00	01	81	80	80	11	BE	7C	B8	:83
C960:	00	00	00	00	00	7F	FF	00	3C	7E	7F	7C	00	00	00	00	:33
C970:	00	C1	83	00	7F	FE	7F	FE	00	00	00	00	00	FF	FF	00	:3C
C980:	F8	FF	FF	0F	00	00	00	00	00	40	02	00	E0	7F	FE	73	:17
C990:	00	00	00	00	00	7F	FE	00	C3	FF	FF	E1	00	00	00	00	:1F
C9A0:	00	00	00	00	41	C0	01	C0	00	00	07	F8	00	00	07	F8	:C0
C9B0:	00	E0	00	00	0F	E0	00	00	0F	E0	00	00	00	00	00	00	:BE
C9C0:	00	00	00	00	C8	00	00	10	C8	00	00	10	00	20	04	00	:D4
C9D0:	38	00	00	B0	38	E0	07	B0	00	10	08	00	08	00	00	60	:37
C9E0:	08	70	0E	60	00	00	00	00	08	01	80	20	08	3E	7C	20	:71
C9F0:	00	00	00	00	08	01	80	10	0A	7E	7E	10	00	00	00	00	:AF
sum :	69	E3	64	BB	0D	C0	D1	ED	FD	3C	13	88	11	05	30	03	:13
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CA00:	04	01	80	18	04	62	46	98	00	00	00	00	02	0D	B0	60	:00
CA10:	0A	62	46	60	00	00	00	00	01	01	81	80	1D	3E	7D	88	:75
CA20:	00	00	00	00	00	FF	FE	00	3E	FE	7E	3C	00	00	00	00	:F3
CA30:	00	C1	83	00	7F	FE	7F	FE	00	00	00	00	00	FF	FF	00	:3C
CA40:	F0	FF	FF	1F	00	00	00	00	00	40	02	00	CE	7F	FE	07	:A1
CA50:	00	00	00	00	00	7F	FE	00	87	FF	FF	C3	00	00	00	00	:C5
CA60:	00	00	00	00	03	80	03	82	1F	E0	00	00	1F	E0	00	00	:06
CA70:	00	00	07	00	00	00	07	F0	00	00	07	F0	00	00	00	00	:F5
CA80:	00	00	00	00	C8	00	00	10	C8	00	00	10	00	20	04	00	:D4
CA90:	38	00	00	B0	38	E0	07	B0	00	10	08	00	08	00	00	60	:37
CAA0:	08	70	0E	60	00	00	00	00	08	01	80	20	08	3E	7C	20	:71
CAB0:	00	00	00	00	08	01	80	10	0A	7E	7E	10	00	00	00	00	:AF
CAC0:	00	00	00	18	06	7F	FE	98	04	00	00	00	00	00	00	00	:37
CAD0:	0F	0F	F0	E0	0E	00	00	E0	00	00	00	00	1F	EA	57	F8	:34
CAE0:	11	E1	87	18	00	00	00	00	3F	FE	7F	FC	20	07	E0	0C	:5C
CAF0:	00	00	00	00	7F	F8	1F	FE	40	01	80	06	00	00	00	00	:5B
sum :	5E	83	D4	B7	21	B6	6F	4E	42	AC	0C	B1	5B	F8	E1	73	:52



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CB00:	FC	7E	7E	3F	40	00	00	02	00	3F	FC	00	EE	3F	FC	0F	:EC
CB10:	00	00	00	02	00	00	00	00	C7	FF	FF	C7	00	00	00	00	:8E
CB20:	00	00	00	00	03	80	03	86	1F	E0	00	00	1F	E0	00	00	:0A
CB30:	00	00	07	00	00	00	07	F0	00	00	07	F0	00	00	00	00	:F5
CB40:	00	00	00	00	08	00	00	13	08	00	00	13	00	20	04	00	:5A
CB50:	0D	00	00	1C	0D	E0	07	1C	00	10	08	00	06	00	00	10	:67
CB60:	06	70	0E	10	00	00	00	00	04	01	80	10	04	3E	7C	10	:F7
CB70:	00	00	00	00	08	01	80	10	08	7E	7E	50	00	00	00	00	:ED
CB80:	18	00	00	00	19	7F	FE	60	00	00	00	20	00	00	00	00	:2E
CB90:	07	0F	F0	F0	07	00	00	70	00	00	00	00	1F	EA	57	F8	:C5
CBA0:	18	E1	87	88	00	00	00	00	3F	FE	7F	FC	30	07	E0	04	:DB
CBB0:	00	00	00	00	7F	F8	1F	FE	60	01	80	02	00	00	00	00	:77
CBC0:	FC	7E	7E	3F	40	00	00	02	00	3F	FC	00	F0	3F	FC	77	:56
CBD0:	40	00	00	00	00	00	00	00	E3	FF	FF	E3	00	00	00	00	:04
CBE0:	00	00	00	00	61	C0	01	C0	00	00	07	F8	00	00	07	F8	:E0
CBF0:	00	E0	00	00	0F	E0	00	00	0F	E0	00	00	00	00	00	00	:BE
sum :	82	3C	88	24	AF	78	AF	47	8B	CA	09	23	56	AD	B6	9A	:5B

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CC00:	00	01	FF	00	00	01	FF	00	00	00	00	00	0F	FF	F0	:FE	
CC10:	00	0F	FF	F0	00	00	00	00	00	3F	FF	FC	00	3F	FF	FC	:72
CC20:	00	30	00	00	00	1F	FF	FE	00	FF	FF	FE	00	18	01	F8	:59
CC30:	01	FF	FF	FF	01	FF	FF	0F	01	FC	07	0C	00	07	FF	FF	:21
CC40:	03	FF	FF	0F	00	06	07	0C	07	3F	FF	FE	07	3F	FF	FE	:AF
CC50:	07	3F	81	F8	00	00	1F	FC	0C	F8	1F	FC	00	00	00	00	:F9
CC60:	19	E0	0F	F8	19	E1	8F	F8	19	E1	8F	F8	00	00	00	00	:02
CC70:	39	FF	CF	E2	00	07	C0	02	3C	78	0F	C0	3C	7F	CF	CF	:8E
CC80:	3C	7F	CF	CF	00	00	00	00	3F	00	1F	8F	00	00	00	0F	:55
CC90:	1F	FF	FE	00	1F	FF	FE	00	1F	FF	FE	00	E0	00	00	60	:94
CCA0:	1F	FF	F8	00	E0	00	00	60	F8	00	1F	C0	00	00	00	00	:2D
CCB0:	F8	00	1F	C0	3F	80	00	00	00	00	00	00	3F	80	00	00	:55
CCC0:	00	FF	80	00	00	FF	80	00	00	00	00	00	0F	FF	F0	00	:FC
CCD0:	0F	FF	F0	00	00	00	00	00	3F	FF	FC	00	3F	FF	FC	00	:72
CCE0:	00	00	0C	00	7F	FF	F8	00	7F	FF	FF	00	1F	80	18	00	:B6
CCF0:	FF	FF	FF	80	F0	FF	FF	80	30	E0	3F	80	FF	FF	E0	00	:98
sum :	DD	D6	BA	DF	C7	89	E7	EF	AD	A7	37	87	CE	28	B0	1F	:49

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CD00:	F0	FF	FF	C0	30	E0	60	00	7F	FF	FC	E0	7F	FF	FC	E0	:D2
CD10:	1F	81	FC	E0	3F	F8	00	00	3F	F8	1F	30	00	00	00	00	:39
CD20:	1F	F0	07	98	1F	F1	87	98	1F	F1	87	98	00	00	00	00	:0C
CD30:	47	F3	FF	9C	40	03	E0	00	03	F0	1E	3C	F3	F3	FE	3C	:65
CD40:	F3	F3	FE	3C	00	00	00	00	F1	F8	00	FC	F0	00	00	00	:F5
CD50:	00	7F	FF	F8	00	7F	FF	F8	00	7F	FF	F8	06	00	00	07	:6F
CD60:	00	1F	FF	F8	06	00	00	07	03	F8	00	1F	00	00	00	00	:3D
CD70:	03	F8	00	1F	00	00	01	FC	00	00	00	00	00	00	01	FC	:14
CD80:	00	01	FF	00	00	01	FF	00	00	00	00	00	00	0F	FF	F0	:FE
CD90:	00	0F	FF	F0	00	00	00	00	00	3F	FF	FC	00	3F	FF	FC	:72
CDA0:	00	30	00	00	00	1F	FF	FE	00	FF	FF	FE	00	18	01	F8	:59
CDB0:	01	FF	FF	FF	01	FF	FF	0F	01	FC	07	FC	00	07	FF	FF	:11
CDC0:	03	FF	FF	0F	00	06	07	FC	07	3F	FF	FE	07	3F	FF	FE	:9F
CDD0:	07	3F	81	F8	00	00	7F	FC	0E	7F	FF	FC	00	00	60	00	:22
CDE0:	1E	78	1F	F8	1E	78	1F	F8	1E	78	1F	F8	00	00	00	00	:07
CDF0:	3E	79	8F	E3	00	01	80	03	3F	38	0F	C0	3F	3F	CF	CF	:0F
sum :	D2	5A	28	F0	F3	E9	E9	93	47	EF	F0	9F	AE	DD	27	CF	:E2

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CE00:	3F	3F	CF	CF	00	00	00	00	3F	9F	9F	80	00	07	80	00	:A0
CE10:	1F	C0	3E	00	1F	C0	3E	00	1F	C0	3E	00	00	00	00	00	:57
CE20:	07	FF	F8	00	00	00	00	00	0E	1F	18	00	00	1F	00	00	:62



```

CE30: 0E 1F 18 00 03 C3 F0 00 00 00 00 00 03 C3 F0 00 :B1
CE40: 00 FF 80 00 00 FF 80 00 00 00 00 00 0F FF F0 00 :FC
CE50: 0F FF F0 00 00 00 00 00 3F FF FC 00 3F FF FC 00 :72
CE60: 00 00 0C 00 7F FF F8 00 7F FF FF 00 1F 80 18 00 :B6
CE70: FF FF FF 80 F0 FF FF 80 3F E0 3F 80 FF FF E0 00 :A7
CE80: F0 FF FF C0 3F E0 60 00 7F FF FC E0 7F FF FC E0 :E1
CE90: 1F 81 FC E0 3F FE 00 00 3F FF FE 70 00 06 00 00 :6B
CEA0: 1F F8 1E 78 1F F8 1E 78 1F F8 1E 78 00 00 00 00 :07
CEB0: C7 F1 9E 7C C0 01 80 00 03 F0 1C FC F3 F3 FC FC :FC
CEC0: F3 F3 FC FC 00 00 00 00 01 F9 F9 FC 00 01 E0 00 :AE
CED0: 00 7C 03 F8 00 7C 03 F8 00 7C 03 F8 00 00 00 00 :65
CEE0: 00 1F FF E0 00 00 00 00 00 18 F8 70 00 00 F8 00 :76
CEF0: 00 18 F8 70 00 0F C3 C0 00 00 00 00 00 0F C3 C0 :A4
-----
sum : 69 29 45 27 EE E2 69 B0 4A CF 57 28 E1 6E E7 9C :51

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: 00 0F F0 00 00 0F F0 00 00 00 00 00 00 7F FE 00 :7B
CF10: 00 7F FE 00 00 00 00 00 01 FF FF 80 01 FF FF 80 :7B
CF20: 00 00 00 00 03 FF FF C0 03 FF FF C0 00 07 E0 00 :69
CF30: 07 FF FF E0 07 FC 3F E0 00 1C 38 00 07 FF FF E0 :40
CF40: 07 FC 3F E0 00 1C 38 00 07 FF FF E0 07 FF FF E0 :40
CF50: 06 07 E0 60 01 FF FF 80 1F FF FF F8 01 C0 03 80 :25
CF60: 3F FF FE 1C 3F FF FE 1C 3F FF FE 1C 00 00 00 00 :08
CF70: 61 FF F8 CE 00 00 00 C0 C0 FF F8 0F D8 FF F9 FF :7B
CF80: D8 FF F9 FF 00 00 00 00 FC FF F9 FE 3C 00 01 F0 :EE
CF90: 40 FF FC 00 7C FF FC 00 7C FF FC 00 00 00 00 00 :29
CFA0: 01 FF FF C0 00 00 00 00 0F CF FF 80 00 0F FF 80 :AA
CFB0: 0F CF FF 80 03 E0 01 F0 00 00 00 00 03 E0 01 F0 :05
CFC0: 00 0F F0 00 00 0F F0 00 00 00 00 00 00 7F FE 00 :7B
CFD0: 00 7F FE 00 00 00 00 00 01 FF FF 80 01 FF FF 80 :7B
CFE0: 00 00 00 00 03 FF FF C0 03 FF FF C0 00 07 E0 00 :69
CFF0: 07 FF FF E0 07 FC 3F E0 00 1F F8 00 07 FF FF E0 :03
-----
sum : E3 E7 E2 29 D3 0D 8E 8C B4 00 14 01 2F B5 B4 7F :AF

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D000: 07 FC 3F E0 00 1F F8 00 07 FF FF E0 07 FF FF E0 :03
D010: 06 07 E0 60 01 FF FF 80 1F FF FF F8 01 C0 03 80 :25
D020: 38 7F FF FC 38 7F FF FC 38 7F FF FC 00 00 00 00 :16
D030: 73 1F FF 86 03 00 00 00 F0 1F FF 03 FF 9F FF 1B :E3
D040: FF 9F FF 1B 00 00 00 00 7F 9F FF 3F 0F 80 00 3C :DF
D050: 00 3F FF 02 00 3F FF 3E 00 3F FF 3E 00 00 00 00 :38
D060: 03 FF FF 80 00 00 00 00 01 FF F3 F0 01 FF F0 00 :54
D070: 01 FF F3 F0 0F 80 07 C0 00 00 00 00 0F 80 07 C0 :8F
D080: 00 0F F0 00 00 0F F0 00 00 00 00 00 00 7F FE 00 :7B
D090: 00 7F FE 00 00 00 00 00 01 FF FF 80 01 FF FF 80 :7B
DOA0: 00 00 00 00 03 FF FF C0 03 FF FF C0 00 00 00 00 :82
DOB0: 07 FF FF E0 07 FF FF E0 00 00 00 00 07 FF FF E0 :AF
DOC0: 07 FF FF E0 00 FF FF 00 07 FF FF E0 07 FF FF E0 :AD
DOD0: 07 FF FF E0 00 00 00 00 1F FB DF F8 00 00 00 00 :D6
DOE0: 3F FB DF FC 3F FB DF FC 3F FB DF FC 00 00 00 00 :3F
DOF0: 67 FB DF E6 00 00 00 00 E7 FB DF E7 E7 FB DF E7 :77
-----
sum : 76 FE B6 D1 94 63 C8 16 1E 67 87 3F 1C 04 D2 9E :7B

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D100: E7 FB DF E7 00 00 00 00 67 FB DF E7 00 00 00 00 :D0
D110: 07 FB DF E0 07 FB DF E6 07 FB DF E6 00 00 00 00 :4F
D120: 03 FF FF C0 00 00 00 00 01 FF E7 E0 01 FF E0 00 :68
D130: 01 FF E7 E0 0F 80 07 F0 00 00 00 00 0F 80 07 F0 :D3
D140: 00 0F F0 00 00 0F F0 00 00 00 00 00 00 7F FE 00 :7B
D150: 00 7F FE 00 00 00 00 00 01 FF FF 80 01 FF FF 80 :7B
D160: 00 00 00 00 03 FF FF C0 03 FF FF C0 00 00 00 00 :82
D170: 07 FF FF E0 07 FF FF E0 00 00 00 00 07 FF FF E0 :AF

```



```

D180: 07 FF FF E0 00 FF FF 00 07 FF FF E0 07 FF FF E0 :AD
D190: 07 FF FF E0 00 00 00 00 1F FB DF F8 00 00 00 00 :D6
D1A0: 3F FB DF FC 3F FB DF FC 3F FB DF FC 00 00 00 00 :3F
D1B0: 67 FB DF E6 00 00 00 00 E7 FB DF E7 E7 FB DF E7 :77
D1C0: E7 FB DF E7 00 00 00 00 E7 FB DF E6 00 00 00 00 :4F
D1D0: 07 FB DF E0 67 FB DF E0 67 FB DF E0 00 00 00 00 :03
D1E0: 03 FF FF C0 00 00 00 00 07 E7 FF 80 00 07 FF 80 :B4
D1F0: 07 E7 FF 80 0F E0 01 F0 00 00 00 00 0F E0 01 F0 :2D
-----
sum : A5 51 09 F0 D5 5D 92 42 14 C0 FC EE 15 DD C1 87 :ED

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D200: 00 00 00 00 00 0F F0 00 00 0F F0 00 00 00 00 00 :FE
D210: 00 78 7E 00 00 78 7E 00 00 00 00 00 01 F0 3F 80 :9C
D220: 01 F0 3F 80 00 00 00 00 03 F8 7F C0 03 F8 7F C0 :24
D230: 00 00 00 00 03 9F FF E0 03 87 FF E0 00 00 00 00 :EA
D240: 01 CF FE 00 01 C0 00 00 00 00 00 00 00 00 79 FC 00 :04
D250: 00 78 00 00 00 00 00 00 00 1F FE 00 00 1F FE 00 :B2
D260: 0F 8F E0 00 00 00 00 08 0F 8F E0 08 38 F0 30 00 :64
D270: 00 00 00 19 38 F0 30 19 00 1F 18 10 78 00 00 0E :57
D280: 78 1F 18 1E 00 E0 3F 18 E4 00 00 00 E4 E0 3F 18 :03
D290: 01 FF FC E0 A0 00 00 00 A1 FF FC E0 01 FF F8 00 :F0
D2A0: 00 00 00 00 01 FF F8 00 03 01 E0 00 00 00 00 00 :DC
D2B0: 0F 01 E7 C0 00 00 00 00 00 00 00 00 07 C0 3E 00 :BC
D2C0: 00 00 00 00 00 0F F0 00 00 0F F0 00 00 00 00 00 :FE
D2D0: 00 7E 1E 00 00 7E 1E 00 00 00 00 00 01 FC 0F 80 :C4
D2E0: 01 FC 0F 80 00 00 00 00 03 FE 1F C0 03 FE 1F C0 :4C
D2F0: 00 00 00 00 07 FF F9 C0 07 FF E1 C0 00 00 00 00 :66
-----
sum : 9A D7 C3 D7 E4 41 DB D9 A7 67 30 18 A4 09 8B A6 :18

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D300: 00 7F F3 80 00 00 03 80 00 00 00 00 00 3F 9E 00 :52
D310: 00 00 1E 00 00 00 00 00 00 7F F8 00 00 7F F8 00 :0C
D320: 00 07 F1 F0 10 00 00 00 10 07 F1 F0 00 0C 0F 1C :27
D330: 98 00 00 00 98 0C 0F 1C 08 18 F8 00 70 00 00 1E :0D
D340: 78 18 F8 1E 18 FC 07 00 00 00 00 27 18 FC 07 27 :2A
D350: 07 3F FF 80 00 00 00 05 07 3F FF 85 00 1F FF 80 :32
D360: 00 00 00 00 00 1F FF 80 00 07 80 C0 00 00 00 00 :E5
D370: 03 E7 80 F0 00 00 00 00 00 00 00 00 00 7C 03 E0 :B9
D380: 00 00 00 00 00 0F F0 00 00 0F F0 00 00 00 00 00 :FE
D390: 00 78 7E 00 00 78 7E 00 00 00 00 00 01 F0 3F 80 :9C
D3A0: 01 F0 3F 80 00 00 00 00 03 F8 7F C0 03 F8 7F C0 :24
D3B0: 00 00 00 00 03 9F FF E0 03 9F FF E0 00 00 00 00 :02
D3C0: 01 CF FF E0 01 C7 FF E0 00 00 00 00 00 77 FC 00 :C9
D3D0: 00 70 00 00 00 00 00 00 00 1F FE 00 00 1F FE 00 :AA
D3E0: 00 0F E0 00 00 00 00 18 00 0F E0 18 07 E0 30 00 :25
D3F0: 00 00 00 32 07 E0 30 32 18 1F 18 60 00 00 00 1C :46
-----
sum : 1C 7A 15 90 CB F4 B4 2B 3D D7 C4 74 93 BF 96 1D :2A

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D400: 18 1F 18 7C 00 C0 3F C0 30 00 00 00 30 C0 3F C0 :A9
D410: 01 FF FC 00 68 00 00 00 69 FF FC 00 01 FF F8 00 :C0
D420: 20 00 00 00 21 FF F8 00 00 3F 80 00 00 00 00 00 :F7
D430: 04 3F 80 00 00 00 00 00 00 00 00 00 03 0F F0 00 :C5
D440: 00 00 00 00 00 0F F0 00 00 0F F0 00 00 00 00 00 :FE
D450: 00 7E 1E 00 00 7E 1E 00 00 00 00 00 01 FC 0F 80 :C4
D460: 01 FC 0F 80 00 00 00 00 03 FE 1F C0 03 FE 1F C0 :4C
D470: 00 00 00 00 07 FF F9 C0 07 FF F9 C0 00 00 00 00 :7E
D480: 07 FF F3 80 07 FF E3 80 00 00 00 00 00 3F EE 00 :0F
D490: 00 00 0E 00 00 00 00 00 00 7F F8 00 00 7F F8 00 :FC
D4A0: 00 07 F0 00 18 00 00 00 18 07 F0 00 00 0C 07 E0 :11
D4B0: 4C 00 00 00 4C 0C 07 E0 06 18 F8 18 38 00 00 00 :F1
D4C0: 3E 18 F8 18 03 FC 03 00 00 00 00 0C 03 FC 03 0C :82

```



D4D0:	00	3F	FF	80	00	00	00	16	00	3F	FF	96	00	1F	FF	80	:46
D4E0:	00	00	00	04	00	1F	FF	84	00	01	FC	00	00	00	00	00	:A3
D4F0:	00	01	FC	20	00	00	00	00	00	00	00	00	00	0F	F0	C0	:DC
sum :	CF	35	A5	38	FE	71	2A	7A	C1	28	5F	3A	73	BC	34	2C	:05
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D500:	00	00	00	00	00	3F	FC	00	00	3F	FC	00	00	00	00	00	:76
D510:	00	E7	E7	00	00	E7	E7	00	00	00	00	00	03	C3	C3	C0	:E5
D520:	03	C3	C3	C0	00	00	00	00	07	E7	E7	E0	07	E7	E7	E0	:B3
D530:	00	00	00	00	06	7F	FE	60	06	7F	FE	60	00	00	00	00	:C6
D540:	07	1F	F8	E0	07	1F	F8	E0	00	00	00	00	03	CF	F3	C0	:81
D550:	03	C0	03	C0	00	00	00	00	00	FF	FF	00	00	FF	FF	00	:82
D560:	00	1F	F8	00	00	00	00	00	00	1E	78	00	00	3F	FF	E0	:CB
D570:	70	00	00	00	70	30	0F	E0	01	FF	FC	18	70	00	00	00	:83
D580:	71	FE	7C	18	1E	7F	FE	00	00	00	00	0C	1E	70	0E	0C	:52
D590:	00	FF	FF	00	00	00	00	1C	00	FE	7F	1C	00	03	FC	00	:B2
D5A0:	00	00	00	00	00	03	FC	00	00	01	F8	00	00	00	00	00	:F8
D5B0:	01	F9	F9	80	00	00	00	00	00	00	00	00	00	F0	07	00	:6A
D5C0:	00	00	00	00	00	3F	FC	00	00	3F	FC	00	00	00	00	00	:76
D5D0:	00	F3	E7	00	00	F3	E7	00	00	00	00	00	03	E1	C3	C0	:1B
D5E0:	03	E1	C3	C0	00	00	00	00	07	F3	E7	E0	07	F3	E7	E0	:E9
D5F0:	00	00	00	00	06	7F	FE	60	06	7F	FE	60	00	00	00	00	:C6
sum :	F2	72	BB	B8	A1	27	C3	9C	1B	71	AC	C0	A5	EE	56	EC	:CB
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D600:	07	3F	FC	E0	07	00	00	E0	00	00	00	00	03	CF	F3	C0	:8E
D610:	03	C0	03	C0	00	00	00	00	00	FF	FF	00	00	FF	FF	00	:82
D620:	00	1F	F8	00	00	00	00	00	00	1E	78	00	07	FF	FC	00	:AF
D630:	00	00	00	0E	07	F0	0C	0E	18	3F	FF	80	00	00	00	0E	:03
D640:	18	3E	7F	8E	00	7F	FE	78	30	00	00	00	30	70	0E	78	:AE
D650:	00	FF	FF	00	38	00	00	00	38	FE	7F	00	00	3F	C0	00	:EA
D660:	00	00	00	00	00	3F	C0	00	00	1F	80	00	00	00	00	00	:9E
D670:	01	9F	9F	80	00	00	00	00	00	00	00	00	00	E0	0F	00	:AE
D680:	00	00	00	00	00	3F	FC	00	00	3F	FC	00	00	1F	00	00	:95
D690:	00	FF	FF	00	00	FF	FF	00	00	78	00	00	03	FF	FF	C0	:35
D6A0:	03	FF	FF	C0	00	E0	00	00	07	FF	FF	E0	07	FF	FF	E0	:6B
D6B0:	00	E0	00	00	07	FF	FF	E0	07	FF	FF	E0	00	E0	00	00	:8A
D6C0:	07	FF	FF	E0	07	FF	FF	E0	00	78	00	00	03	FF	FF	C0	:03
D6D0:	03	FF	FF	C0	00	00	00	00	00	FF	FF	00	00	FF	FF	00	:BD
D6E0:	00	1E	78	00	00	00	00	00	00	1E	78	00	07	FE	7C	00	:AD
D6F0:	00	00	00	0E	07	FE	7C	0E	18	3E	7F	80	00	00	00	0E	:00
sum :	30	F4	88	2A	5B	C8	3F	34	A6	01	65	C0	4E	55	43	B4	:D2
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D700:	18	3E	7F	8E	00	76	6E	78	30	00	00	00	30	76	6E	78	:7B
D710:	00	EA	57	00	38	00	00	00	38	EA	57	00	00	3F	E0	00	:11
D720:	00	00	00	00	00	3F	E0	00	00	1F	C0	00	00	00	00	00	:FE
D730:	01	9F	CF	00	00	00	00	00	00	00	00	00	00	E0	1F	80	:EE
D740:	00	00	00	00	00	3F	FC	00	00	3F	FC	00	00	1F	00	00	:95
D750:	00	FF	FF	00	00	FF	FF	00	00	78	0E	00	03	FF	FF	C0	:43
D760:	03	FF	FF	C0	00	E0	03	80	07	FF	FF	E0	07	FF	FF	E0	:EE
D770:	00	E0	00	00	07	FF	FF	E0	07	FF	FF	E0	00	E0	00	00	:8A
D780:	07	FF	FF	E0	07	FF	FF	E0	00	78	00	00	03	FF	FF	C0	:03
D790:	03	FF	FF	C0	00	00	00	00	00	FF	FF	00	00	FF	FF	00	:BD
D7A0:	00	1E	78	00	00	00	00	00	00	1E	78	00	00	3E	7F	E0	:C9
D7B0:	70	00	00	00	70	3E	7F	E0	01	FE	7C	18	70	00	00	00	:80
D7C0:	71	FE	7C	18	1E	76	6E	00	00	00	00	0C	1E	76	6E	0C	:1F
D7D0:	00	EA	57	00	00	00	00	1C	00	EA	57	1C	00	07	FC	00	:BD
D7E0:	00	00	00	00	00	07	FC	00	00	03	F8	00	00	00	00	00	:FE
D7F0:	00	F3	F9	80	00	00	00	00	00	00	00	00	01	F8	07	00	:6C
sum :	07	9C	E5	86	D4	8C	33	B4	77	3E	61	00	CC	43	59	44	:17



まず、キャラクタを作らない事にはこれ以後の実験はできません。ぜひみなさんに作っていただきたいのですが、サンプルとして、リスト2-2-3bに、以前私の作った“アストロノートと宇宙人”のキャラクタデータをのせてあります。

リスト最初の &HBF00～&HBF35 までは全キャラクタを画面に表示するルーチンです。&HBFA0～&HBFFF までがリスト2-2-3a のオブジェクトです。

もし、データを打ち込まれましたら、

h) GBF00 

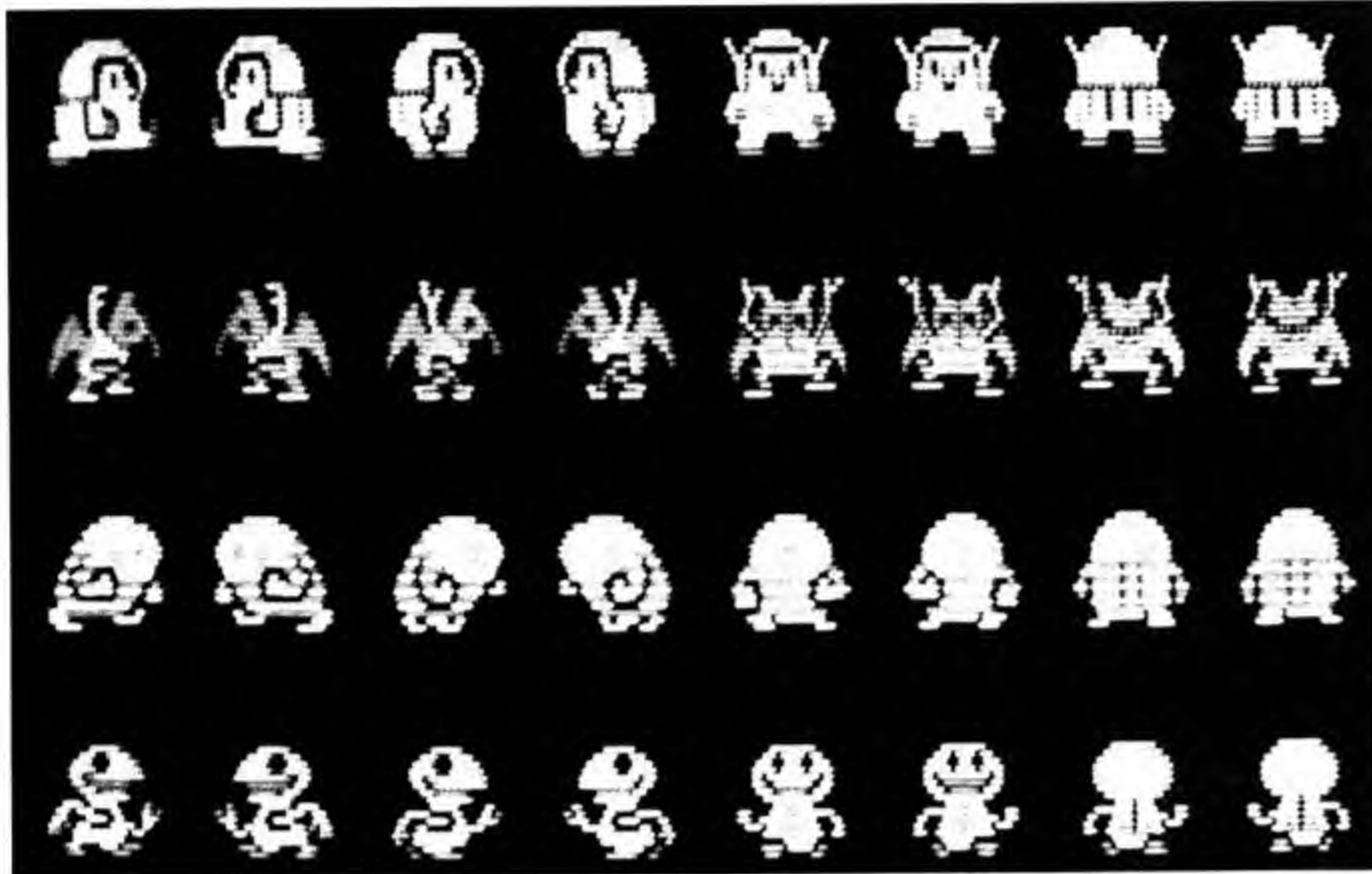


写真2-2-1 キャラクタ表示

として、32個のキャラクタが表示されるのを確認の上、

BSAVE “ASTRO”, &HC000, &H1800 として、ディスクにセーブして下さい。

なお、キャラクタの名称と先頭アドレスを第2-2-1表に示します。

リスト2-2-3はグラフィックRAMの裏にあるメインRAM(&HC000～&HFFFF番地の事)にもキャラクタデータを置く事ができますが、PUTの速さをさらに速くしたい時は、リスト2-2-4を使います。

第2-2-1表 キャラクタデータ先頭アドレス

	右1	左1	右2	左2	前1	前2	後1	後2
アストロノート	C000	C0C0	C180	C240	C300	C3C0	C480	C540
おけさ星人	C600	C6C0	C780	C840	C900	C9C0	CA80	CB40
ラグビー星人	CC00	CCC0	CD80	CE40	CF00	CFC0	D080	D140
キチン星人	D200	D2C0	D380	D440	D500	D5C0	D680	D740



ブロック転送命令で各ブロック RAM にデータを移しています。ただ問題なのはブロック転送のため、キャラクタデータの置く場所を &HBEFF 以前にしなければなりません。しかし、DISK BASIC を使用していると、フリーエリアが大変少ないので、キャラクタデータを BASIC テキスト用の RAM (&H0000 ~ &H7FFF) に移します。

例として、データを &H2000 に移すには、


**BLOAD "ASTRO", &H2000** 

で、&H2000 からロードされます。確実にロードされたかどうかを見るには、アウトポートの &H70 にデータを送ってやります。それは、普通 &H0000 から &H7FFF 番地までは BASIC の ROM になっており、モニタで、

h]E2000 

とやっても、ROM の &H2000 番地が見られるだけでメイン RAM の &H2000 番地を見る事はできません。メイン RAM の &H0000 から &H7FFF を見るためには、&H8000 番地から &H83FF 番地までの、テキストウィンドウと呼ばれる所を見るとメイン RAM を見る事ができます。

たった、&H400 バイトで 32K におよぶメイン RAM を見る事ができるというのは不思議な話ですが、アウトポート &H70 に見たい番地の上位 1 バイトを出力すると、その様子がテキストウィンドウに現れます。

h]O70, 20 

と入力したのち、

h]E8000 

とすると、&H2000 番地の内容が &H8000 番地に見る事ができます。

プログラムで &H0000 から &H7FFF までのメイン RAM を利用したい時には、リスト 2-2-4 の始めにある様に、アウトポート &H31 にデータを送ります。ただし、このポートは色々なモードの設定を行うのでやたらなデータを出力するわけには行きません。そこで、

**LD A, (0E6C2H)**

として、ワークエリア &HE6C2 の内容を A レジスタに読み込みます。E6C2 には、ポート &H31 に設定したデータが保存されているのです。次に、

**OR 2**

として、第 1 ビットを ON (1) にします。つまり、アウトポート &H31 に出力するデータの第 1 ビットが ON の時、オール RAM モードになるのです。

**OUT (31H), A**

として、31H ポートに出力します。BASIC に戻す時には、必ず、

**LD A, (0E6C2H)**

**OUT (31H), A**

として、モードを元に戻さねばなりません。

それでは、キャラクタデータを &H2000 からロードして、リスト 2-2-4 をテストプレイしてみましょう。

プログラムを打ち込んだら、&HBFF0 から &HBFF5 までの 6 バイトに、00, 20, 10, 04, 00, C0 と入れて下さい。リストを見ればお分かりかと思いますが、







## リスト2-2-4b 入力用高速表示プログラム

```
BF00 F3 3A C2 E6 F6 02 D3 31 2A F0 BF ED 5B F4 BF ED
BF10 4B F2 BF C5 D5 48 06 00 C5 D3 5C ED B0 D3 5F C1
BF20 D1 D5 C5 D3 5D ED B0 D3 5F C1 D1 D5 D3 5E ED B0
BF30 D3 5F D1 E5 21 50 00 19 EB E1 C1 0D 20 D5 3A C2
BF40 E6 D3 31 FB C9 00 00 00 00 00 00 00 00 00 00
```

BFF0番地→キャラクタデータの番地(2バイト)

BFF2番地→キャラクタの縦ライン数

BFF3番地→キャラクタの横バイト数

BFF4番地→PUTする画面番地(2バイト)

です。他のPUTや消去プログラムもこれと同じワークエリアとなっています。


では、準備ができましたら、BASICに戻して、10

```
10 CLEAR, &HBEFF: P=&HBF00: X=&HBFF4
```

```
20 FOR I=0 TO 30
```

```
30 POKE X, I*2: CALL P
```

```
40 NEXT
```

**RUN**  しますと、作ったキャラクタが少し重なってズラッと並びましたね。サンプルデータでは宇宙飛行士の右向き姿が現れます。

&HBFF0番地から2バイトをかえると他のキャラクタも出て来ます。たとえば、26C0番地つまりC0, 26を入れると、“おけさ星人”の左姿が出ます。

私のゲームでのキャラクタPUTは大体これを元にしてしていますが、キャラクタが重なった時、前のキャラクタが削られるのはちょっと気になりますね。

そこで、前のキャラクタと重ね合わせる事を考えてみましょう。従来のPC-88では画面データとキャラクタデータを重ね合わせるのに、Aレジスタを使って、OR命令で重ねるのが普通です。

ところが、PC-88SR以降では、ハードウェア処理により、重ね合わせが簡単になりました。リスト2-2-5を見て下さい。ポート&H32と&H34、&H35が新しく出て来ます。そのかわり、&H5C～&H5Eにはアクセスしていません。第2-2-2表にポート&H32、&H34、&H35の機能一覧があります。これだけではわかりにくいと思いますので、実験しながら見て下さい。

まず、リスト2-2-5のBF13番地に、

```
IN A, (32H)
```

とあって、ポート&H32からデータを取り入れています。このポートは入力が可能なので、今どんなモードになっているか、IN命令で知る事ができるのです。(ポート&H34と&H35は出力のみですからIN命令が使えません)。

そこで“PUSH”により、この値を保存し、

```
OR 40H OUT (32H), A
```

で、第6ビットをONにします。これで、PC-88SR以降の拡張機能が使えるようになりました。

次にポート&H35に&H80を出力します。この意味は、第7ビットが1であとは0ですから、“グラフィックRAMをアクセスする”、“3画面同時ライトを行う”の意味があります。



## リスト2-2-5a 画面データとキャラクタデータの重ね合わせ

	:		
	; ALU OR PUT	IN BFF0H,キャラクター データ アドレス (-BEFFH)	
	:	BFF2H,タテ ライン スク	
	:	BFF3H,ヨコ ハイト スク	
	:	BFF4H,カメン アドレス	
	:		
	ORG 0BF00H		
BF00 F3	APUT: DI		;フリコミ タメ
	:		
BF01 3AC2E6	LD A,(0E6C2H)		
BF04 F602	OR 2		
BF06 D331	OUT (31H),A		;ALL RAM MODE
	:		
BF08 2AF0BF	LD HL,(0BFF0H)		;READ CHR DATA ADDRESS
BF0B ED5BF4BF	LD DE,(0BFF4H)		;READ G RAM ADDRESS
BF0F ED4BF2BF	LD BC,(0BFF2H)		;READ X=B Y=C
	:		
BF13 DB32	IN A,(32H)		
BF15 F5	PUSH AF		
BF16 F640	OR 40H		;KAKUTHIOO MODE
BF18 D332	OUT (32H),A		
	:		
BF1A 3E80	LD A,80H		
BF1C D335	OUT (35H),A		;KAKIKOMI
	:		
BF1E C5	APUTL: PUSH BC		
BF1F 48	LD C,B		;BC=B (X)
BF20 0600	LD B,0		
	:		
BF22 3E67	LD A,67H		;ONLY BLUE PLANE
BF24 D334	OUT (34H),A		;BIT SET MODE
	:		
BF26 D5	PUSH DE		
BF27 C5	PUSH BC		
BF28 EDB0	LDIR		
BF2A C1	POP BC		
BF2B D1	POP DE		
	:		
BF2C 3E57	LD A,57H		;ONLY RED PLANE
BF2E D334	OUT (34H),A		;BIT SET MODE
	:		
BF30 D5	PUSH DE		
BF31 C5	PUSH BC		
BF32 EDB0	LDIR		
BF34 C1	POP BC		
BF35 D1	POP DE		
	:		
BF36 3E37	LD A,37H		;ONLY GREEN
BF38 D334	OUT (34H),A		;BIT SET MODE
	:		
BF3A D5	PUSH DE		
BF3B EDB0	LDIR		
BF3D D1	POP DE		
	:		
BF3E EB	EX DE,HL		
BF3F 015000	LD BC,50H		
BF42 09	ADD HL,BC		;NEXT LINE
BF43 EB	EX DE,HL		
	:		
BF44 C1	POP BC		
BF45 0D	DEC C		
BF46 20D6	JR NZ,APUTL		
	:		
BF48 AF	XOR A		



```

BF49 D335      OUT  (35H),A      ;MAIN RAM アクセス
BF4B F1        POP  AF          ;NORMAL MODE
BF4C D332      OUT  (32H),A
BF4E D35F      OUT  (5FH),A      ;RAM
;
BF50 3AC2E6    LD   A,(0E6C2H)
BF53 D331      OUT  (31H),A
BF55 FB        EI
BF56 C9        RET
;
;
BF57           END

```

## リスト2-2-5b 入力用プログラム

```

BF00 F3 3A C2 E6 F6 02 D3 31 2A F0 BF ED 5B F4 BF ED
BF10 4B F2 BF DB 32 F5 F6 40 D3 32 3E 80 D3 35 C5 48
BF20 06 00 3E 67 D3 34 D5 C5 ED B0 C1 D1 3E 57 D3 34
BF30 D5 C5 ED B0 C1 D1 3E 37 D3 34 D5 ED B0 D1 EB 01
BF40 50 00 09 EB C1 0D 20 D6 AF D3 35 F1 D3 32 D3 5F
BF50 3A C2 E6 D3 31 FB C9 00 00 00 00 00 00 00 00

```

しかし、3画面にデータを同時にライトしたのでは、全部白いキャラクタになってしまいますね。そこで、同時ライトのやり方を決めるのが、ポート&H34です。

リストのBF24番地に、&H67をポート&H34に出力しているのが見られます。

&H67をビットで見ると **0 1 1 0 0 1 1 1** ですね。第2-2-2表を見ると、ビットの組合せが、青のGRAMに対してのみビットセットモードで、他のGRAMには何もしない様になっています。

青のデータをPUTしたいのですから、これでよいわけです。この状態でブロック転送しています。もしポート&H34に&H07を出力すると、青のデータだけで3画面同時にPUTする事ができるのです(すごいですね。もちろん白い色に見えますので色の変化をつけるわけにはいきません)。

次に、ポート&H34に&H57を出力して赤のプレーンにのみデータをPUTし、&H37を出力し

## 第2-2-2表 ポート &amp;H32, &amp;H34, &amp;H35 の機能一覧

## ポート 32H の機能〔各種モード指定〕

ビット	機 能
0 1	内部ROMの選択
2 3	テレビ出力モードの選択
4	SR以降つけられたF000H~FFFFHまでのテキスト表示用メモリの選択 0で専用メモリ, 1でメインメモリを選択
⑤	0でデジタルカラーモード 1でアナログカラーモード
⑥	グラフィックRAM同時アクセスを決める 0で各GRAM別にアクセス, 1で同時アクセス可
7	PSGの割り込み許可を決める 0で割り込み許可, 1で割り込み禁止



## ポート 34H の機能〔グラフィック合成指定〕

ビット	機 能	
①	青 A	A と B のビットの組合せにより各 GRAM に対するデータセットモードを決定します
②	赤 A	
③	緑 A	
3	未使用	
④	青 B	
⑤	赤 B	
⑥	緑 B	
7	未使用	
		B : A
		0 : 0 →ビットリセット (PUT するビットの所が消える)
		0 : 1 →ビットセット (データを OR PUT する)
		1 : 0 →ビット反転 (XOR PUT の様になる)
		1 : 1 →その GRAM に対しては 何もしない。

## ポート 35H の機能〔グラフィック合成指定およびモード〕

ビット	機 能	
0	青	0 ~ 2 のビットをセットすると GRAM からレジスタへ読み込む命令を行った時、ビットがセットされている GRAM 同志を比較して AND をとったものがレジスタに入る。従って三つのうち一つだけビットがセットされていると、そのプレーンのデータのみ読み込まれる
1	赤	
2	緑	
3	未使用	4, 5 ビットは組みで次の動作を指定する
④	A	B : A
⑤	B	0 : 0 →青赤緑の 3 画面にデータを同時ライト 0 : 1 →読み込んだデータを同時ライト、画面データの移動に強力でスクロール等を使用 1 : 0 →赤のプレーンから青のプレーンに移す時使用 1 : 1 →青のプレーンから赤のプレーンに移す時使用
6	未使用	
⑦	0: 1:	メイン RAM アクセス ) 同時書き込み等、新しい機能でグラフィックを読み書きする GRAM アクセス ) 時には必ず 1 にする

※特に重要なビットに○印をつけてあります

て緑のプレーンにのみデータを PUT します。

この PUT は、すでに画面上にセットされているビットは変化させないので、“OR PUT”になるのです。このルーチンをテストするには、リスト 2-2-4 でやったのとまったく同様に準備します。準備が終わったら、前のプログラムで使いましたテスト用の BASIC プログラムを RUN させます。



## リスト2-2-6a キャラクターくり抜きプログラム

```

;
; ALU RESET
;
; (キャラクター クリヌキ)
;
; IN BFF0H,キャラクター データ アドレス (-BDFH)
; BFF2H,タテ ライン スウ
; BFF3H,ヨコ ハイ スウ
; BFF4H,カメン アドレス
;
; ORG 0BE00H
;
BE00 F3 ARESET:DI ;ワリコミ タメ
;
BE01 3AC2E6 LD A,(0E6C2H)
BE04 F602 OR 2
BE06 D331 OUT (31H),A ;ALL RAM MODE
;
BE08 2AF0BF LD HL,(0BFF0H) ;READ CHR DATA ADDRESS
BE0B ED5BF4BF LD DE,(0BFF4H) ;READ G RAM ADDRESS
BE0F ED4BF2BF LD BC,(0BFF2H) ;READ X=B Y=C
;
BE13 DB32 IN A,(32H)
BE15 F5 PUSH AF
BE16 F640 OR 40H ;KAKUTHIOO MODE
BE18 D332 OUT (32H),A
;
BE1A 3E80 LD A,80H
BE1C D335 OUT (35H),A ;KAKIKOMI
BE1E AF XOR A
BE1F D334 OUT (34H),A ;BIT RESET MODE
;
BE21 C5 ARESL1:PUSH BC
;
BE22 3E03 LD A,3 ;COUNTER SET
;
BE24 48 LD C,B ;BC=B (X)
BE25 0600 LD B,0
;
BE27 D5 ARESL2:PUSH DE
BE28 C5 PUSH BC
BE29 EDB0 LDIR
BE2B C1 POP BC
BE2C D1 POP DE
BE2D 3D DEC A
BE2E 20F7 JR NZ,ARESL2
;
BE30 EB EX DE,HL
BE31 015000 LD BC,50H
BE34 09 ADD HL,BC ;NEXT LINE
BE35 EB EX DE,HL
;
BE36 C1 POP BC
BE37 0D DEC C
BE38 20E7 JR NZ,ARESL1
;
BE3A AF XOR A
BE3B D335 OUT (35H),A ;MAIN RAM
BE3D F1 POP AF ;NORMAL MODE
BE3E D332 OUT (32H),A
BE40 D35F OUT (5FH),A ;RAM
;
BE42 3AC2E6 LD A,(0E6C2H)
BE45 D331 OUT (31H),A
;
BE47 FB EI ;ワリコミ OK!
BE48 C9 RET
;
;
BE49 END

```



## リスト2-2-6b 入力用プログラム

```

BE00 F3 3A C2 E6 F6 02 D3 31 2A F0 BF ED 5B F4 BF ED
BE10 4B F2 BF DB 32 F5 F6 40 D3 32 3E 80 D3 35 AF D3
BE20 34 C5 3E 03 48 06 00 D5 C5 ED B0 C1 D1 3D 20 F7
BE30 EB 01 50 00 09 EB C1 0D 20 E7 AF D3 35 F1 D3 32
BE40 D3 5F 3A C2 E6 D3 31 FB C9 00 00 00 00 00 00

```

どうです？ ずらっと並んだキャラクタが重なり合っていますね。でも少し変です。重なった所が白っぽく透けたようになっています。

“OR PUT”ではこうなるのが当り前ですが、何とかしたいものです。リスト2-2-4ではキャラクタの回りが四角く消えてしまいましたが、理想的には、キャラクタの所だけ消して、それからキャラクタを書けばよいのです。

その方法を次に紹介いたしますが、その前に一言、リスト2-2-5のBF49番地以後のOUT命令はモードを元に戻すもので、これをやらないと書いたあと暴走したり、BASICの“CLS 2”でおかしくなったりします。

さて、リスト2-2-6はキャラクタの所だけ画面を消すプログラムです。内容はほとんどリスト2-2-5と同じですが、ポートの&H34へ0を出力してビットリセットモードにしています。これを3画面に対して行うために3回ループしています。3画面に出力するデータを“OR”してから、3画面同時に書き込むよりもやや高速です(大きいキャラクタだと違いがわかる)。

このプログラムをテストするためには、リスト2-2-5が必要です。画面にキャラクタをPUTする準備ができたなら、次のBASICプログラムを走らせてください。

```

10 CLEAR, &HBDFE
20 P=&HBF00:C=&HBE00:X=&HBFF4
30 FOR I=0 TO 30
40 POKE X, I*2
50 CALL C:CALL P
60 NEXT

```

これで並んだキャラクタがうまく重なったと思います。ただし、キャラクタの中に“黒”があるとそこはバックの絵が出てしまいますので、“黒”は使わないか、あるいはパレットを変えて“黒”にします。

## IV キャラクタを消す

次にキャラクタを消す方法です。

ちょっと変わった方法を先に紹介しましょう。リスト2-2-5で、BF23番地の67を“76”に、BF2D番地の57を“75”に、BF37番地の37を“73”にそれぞれ書き替えます。ビットセットモードからビット反転モードにしたのです。



## リスト2-2-7a キャラクタ消去プログラム


```

;
; ALU CLS
;
; (キャラクター クリアー)
;
; IN BFF2H,タテ ライン スウ
; BFF3H,ヨコ ハイト スウ
; BFF4H,カメン アドレス
;
; ORG 0BE80H
;
BE80 F3 CRESET:DI ;フリコミ タメ
;
BE81 3AC2E6 LD A,(0E6C2H)
BE84 F602 OR 2
BE86 D331 OUT (31H),A ;ALL RAM MODE
;
BE88 2AF4BF LD HL,(0BFF4H) ;READ G RAM ADDRESS
BE8B ED4BF2BF LD BC,(0BFF2H) ;READ X=B Y=C
;
BE8F DB32 IN A,(32H)
BE91 F5 PUSH AF
BE92 F640 OR 40H ;KAKUTHIOO MODE
BE94 D332 OUT (32H),A
;
BE96 3E80 LD A,80H
BE98 D335 OUT (35H),A ;KAKIKOMI
BE9A AF XOR A
BE9B D334 OUT (34H),A ;BIT RESET MODE
;
BE9D C5 CRESL1:PUSH BC
BE9E E5 PUSH HL
BE9F 3EFF LD A,0FFH ;FULL BIT RESET
BEA1 77 CRESL2:LD (HL),A
BEA2 23 INC HL
BEA3 10FC DJNZ CRESL2
BEA5 E1 POP HL
BEA6 015000 LD BC,50H
BEA9 09 ADD HL,BC ;NEXT LINE
BEAA C1 POP BC
BEAB 0D DEC C
BEAC 20EF JR NZ,CRESL1
;
BEAE AF XOR A
BEAF D335 OUT (35H),A ;MAIN RAM
BEB1 F1 POP AF ;NORMAL MODE
BEB2 D332 OUT (32H),A
BEB4 D35F OUT (5FH),A ;RAM
;
BEB6 3AC2E6 LD A,(0E6C2H)
BEB9 D331 OUT (31H),A
BEBB FB EI ;フリコミ OK!
BEBC C9 RET
;
;
BEBD END

```

BFF0からBFF5までの6バイトの設定を前と同様にやり、&H2000からキャラクタデータをロードして、

CLEAR, &HBDFF 


CLS 2 




## リスト2-2-7b 入力用プログラム

BE80	F3	3A	C2	E6	F6	02	D3	31	2A	F4	BF	ED	4B	F2	BF	DB
BE90	32	F5	F6	40	D3	32	3E	80	D3	35	AF	D3	34	C5	E5	3E
BEA0	FF	77	23	10	FC	E1	01	50	00	09	C1	0D	20	EF	AF	D3
BEB0	35	F1	D3	32	D3	5F	3A	C2	E6	D3	31	FB	C9	00	00	00

としたのち、モニタから、

h)GBF00 

とやってみて下さい。画面左上にキャラクタが現れます。続けて、

h)GBF00 

と、同じ様にPUT ルーチンを実行します。するとどうでしょう、前にあったキャラクタが消えてしまいました。

リスト2-2-5を書き替えたこのプログラムは、“XOR PUT”ルーチンです。画面に何もない状態で、XORをすると、データはそのままの状態です。すでに画面に同じものがあると、同じデータ同志のXORは“0”になるので消えてしまうのです。

動きのあるプログラムでは、一般にこの方法はもちいられませんが、この方法で消すと重なりあった部分が簡単に元に戻ります。キャラクタを重ね合わせて試して下さい。

一般に使われるのはリスト2-2-7にあるような、無条件にある大きさの長方形内を消してしまう方法です。

ブロック転送は使っていませんが、青、赤、緑の3プレーン同時に消してしまうのでPC-88従来機より高速です。

なお、ブロック転送によりメモリをクリア(0)にする方法は次の通りです。

```
LD HL, クリアしたい先頭番地
LD DE, クリアしたい先頭番地+1
LD BC, クリアしたいメモリ長-1
LD (HL), 0 (先頭メモリをクリア)
LDIR (ブロック転送)
```

では、クリアルーチンを使ってキャラクタを移動させてみましょう。

キャラクタが表示できるように準備をして、リスト2-2-7を入力しましたら、

```
10 CLEAR, &HBDFF
20 P=&HBF100:C=&HBE80:X=&HBFF4
30 FOR I=0 TO 30
40 POKE X, I*2:CALL P
50 FOR J=0 TO 150:NEXT
60 CALL C:NEXT
```

これをRUNすると、画面の上をキャラクタがすべるように移動します。50行のJの範囲を変えると移動の早さが変わります。

以上、いくつかのPUT、クリアルーチンを紹介してきましたが、これらはワークエリアのBFF0~BFF5番地に値を設定して働きます。これを他の番地へ移すのはプログラムの書き換えが必要です。しかし、プログラム全体は他の番地(&H8400~&HBFEF)へ移動してもかまいません。





## 2-3 パレット・コントロール

PC-88SR 以降は 8 色のデジタルパレットに加えて、アナログパレットが設定できます。これらパレットのコントロールも OUT 命令により実現できます。

始めに、デジタルパレットを使うかアナログパレットを使うか設定する方法を見てみましょう。

### ● デジタルパレットに設定する。

```
IN  A, (32A)   ポート 32H の内容を読む  
AND  0DFH     第 5 ビットを 0 にする  
OUT(32H), A   ポート 32H に出力する
```

### ● アナログパレットに設定する

```
IN  A, (32H)  
OR  20H       第 5 ビットを 1 にする  
OUT(32H), A
```

デジタルでもアナログでもカラーパレットの制御はポートの &H54 ~ &H5B への出力で行います。この八つが元になるパレット番号に対応しています。

デジタルカラーの場合は設定したいカラーコードをこれらのアウトポートに出力すればよいのです。

たとえば、オリジナルの赤を黄色に変えたい時、

COLOR = (2, 6)

BASIC では上の様にしますが、マシン語ではポート &H56 がオリジナルの赤ですから、

```
LD  A, 6       カラーコード 6 (黄)
```

```
OUT(56H), A   パレット 2 に 6 をセット
```

で、よいのです。

COLOR = (7, 1)

をしたい時は、



```
LD  A, 1
OUT(5BH), A
```

とすれば、白が青に変わります。

アナログカラーの場合は、それぞれのアウトポートに、2バイトの数で色を指定します。つまり一つのポートへ2回データを出力する必要があるのです。

アナログカラーでは、青、赤、緑の三原色の輝度を変化させ混ぜ合わせる事ができます。変化の幅は各原色について8段階で、それぞれの値をKB, KR, KGとすると、ポートへ出力させる値は次の式で求まります。

一つ目 =  $KR \times 8 + KB$

二つ目 =  $\&H40 + KG$

(ただし  $0 \leq KB, KR, KG \leq 7$ )

たとえば KB=4, KR=5, KG=2 の割合で色を混ぜ、これを元白のパレットに入れたい時は、

$KR \times 8 + KB = 5 \times 8 + 4 = 44 = \&H2C$

$\&H40 + 2 = \&H42$

だから、

```
LD  A, 2CH
OUT (5BH), A ; パレット 7 に 2CH を出力
```

```
LD  A, 42H
OUT (5BH), A ; パレット 7 に 42H を出力
```

です。これをBASICで実験してみましょう。

```
10 KB=4:KR=5:KG=2
```

```
20 D1=KR*8+KB:D2=&H40+KG
```

```
30 LINE(200, 100)-(300, 150), 7, BF
```

```
40 OUT &H5B, D1:OUT &H5B, D2
```

もちろん、V2モードで実験して下さい。30行で書かれた白の四角形が、40行のOUT命令で着色されます。

最後に、テキスト画面を表示させない方法を紹介します。従来のPC-88ではテキストを表示させておくと全体の処理が遅くなるのでよくこれが使われました。

```
XOR A      A=0
```

```
OUT(51H), A ; ポート 51H に 0 を出力
```

ポート &H51はCRTC(CRTコントローラ)へのデータ入出力です。





## 2-4 フロッピーディスクコントロール

PC-88のフロッピーディスクはそのシステム自体に Z80CPU を用い、PC-88本体からのコマンドによって動作します。PC-88MR/MHでは1メガバイトのフロッピーディスクまでサポートしています。そのため、コマンドがいくつかふえています。もちろんMR/MHでも一般の2Dが使えるので、次に紹介するディスク入力ルーチンや、後で出て来る高速ディスクローダーも使えます。

ディスクと本体とのコマンドやデータのやりとりは、ポート&HFD~&HFFを使います。これらのポートの入出力の実体は省略いたしますが、高速ローダー(第4-2章)の所で少しふれていますので、そちらも参照して下さい。

ここでは、セクタ単位(256バイトずつ)でディスクからデータをメインRAMに移すプログラムを御紹介します。1度のハンドシェイクで2バイトデータを送る形のごく一般的なものですが、オールRAMモードのプログラムを作る際には役に立つと思います(リスト2-4-1)。

使い方は自分で決めたワークエリア内に必要なデータを書き込んで置き、この先頭アドレスをHLレジスタに入れて、&HDE00番地をコールします。

必要なデータは次の通りです。

先頭番地：ドライブNo.-1 (普通は0か1)


- +1：論理トラックNo.
- +2：読み込み開始セクタ (1~16)
- +3：読み込みセクタ数 (1~32)
- +4：ロード先番地下位
- +5：ロード先番地上位

論理トラック番号とは、普通のトラック番号を2倍にして、サーフェース0(表面)なら0を、サーフェース1(裏面)なら1を足します。

サンプルとして、リスト2-4-1bの&HDFA0から6バイトに、00, 00, 01, 02, 00, C0と入っています。



これは、ドライブ1の(00)、トラック0でサーフェース0の(00)、セクタ1から(01)、2セクタだけ(02)、&HC000よりロードせよ、という意味です。

h) GDF80 

を実行すると、ディスクの始めの2セクタ(IPL部分)が,&HC000~&HC1FFまでにロードされるでしょう。値を色々と変えて実験してみてください。

なお、ディスクに書き込む時には、モニタのコントロールWコマンドを使って下さい。使い方はマニュアルをご覧ください。

また、プログラムを置く場所を変える時には、コール先の番地変更等にご注意下さい。

#### リスト2-4-1a ディスクローダー

```

;
;*** DIST LOADER ***
;      IN HL DATA ADRESS TOP
;
;  DATA FORMAT ;; DRIV,TRACK,SECTER,SECTER SU,LOAD ADRES
;
;
;      ORG  0DE00H
;
DE00 7E      RAMPUT:LD  A,(HL)          ;RAMPUT 8006H
DE01 323EDE      LD  (DRIVNO),A
DE04 23          INC  HL
DE05 CD43DE      CALL SETREG
DE08 7E          LD  A,(HL)
DE09 23          INC  HL
DE0A 66          LD  H,(HL)
DE0B 6F          LD  L,A
DE0C 2236DE      LD  (SCTOP),HL        ;RAM ADRES TOP
DE0F CD56DE      CALL DLOAD
DE12 3A41DE      LD  A,(KAZU2)
DE15 67          LD  H,A
DE16 2E00        LD  L,0
DE18 2238DE      LD  (OUTP),HL        ;OUTP=TRANS BIYT
DE1B ED5B36DE    LD  DE,(SCTOP)      ;READ RAM ADRESS
DE1F CD0DDF      RAMPUL:CALL RESDAT   ;RESCEVE DATA  OUT A,L
DE22 12          LD  (DE),A
DE23 13          INC  DE
DE24 7D          LD  A,L
DE25 12          LD  (DE),A
DE26 13          INC  DE
DE27 2A38DE      LD  HL,(OUTP)
DE2A 2B          DEC  HL
DE2B 2B          DEC  HL
DE2C 2238DE      LD  (OUTP),HL
DE2F 7C          LD  A,H              ;KAZU AWASE
DE30 B5          OR   L
DE31 C8          RET  Z
DE32 18EB        JR   RAMPUL         ;GOTO LOOP
;
;
;  * WARK AREA
;
DE34 0000      GTOP:  DW   0
DE36 0000      SCTOP: DW   0
DE38 0000      OUTP:  DW   0
DE3A 0000      XST:   DW   0
DE3C 00        SC:    DB   0
DE3D 00        SECTER:DB  0
DE3E 00        DRIVNO:DB  0
DE3F 00        TRACK: DB  0
DE40 00        KAZU1: DB  0
DE41 00        KAZU2: DB  0
DE42 00        KAZU3: DB  0
;

```



```

***** DISK I/O *****
;
;
; SET WORK AREA
;
DE43 7E      SETREG:LD  A,(HL)
DE44 323FDE      LD  (TRACK),A
DE47 23      INC  HL
DE48 7E      LD  A,(HL)
DE49 323DDE      LD  (SECTER),A
DE4C 23      INC  HL
DE4D 7E      LD  A,(HL)
DE4E 3240DE      LD  (KAZU1),A
DE51 3241DE      LD  (KAZU2),A
DE54 23      INC  HL
DE55 C9      RET

;
; DISK MOVE AND LOAD
;
DE56 3E50      DLOAD:LD  A,50H      ;DISK MEMORY TOP 5000H
DE58 323CDE      LD  (SC),A
DE5B 3A3DDE      DLOAD1:LD  A,(SECTER)
DE5E 4F      LD  C,A
DE5F 3A40DE      LD  A,(KAZU1)
DE62 81      ADD  A,C      ;TOP SCTER+KAZU
DE63 FE12      DLOAD2:CP  18      ;-18
DE65 3803      JR  C,DLOAD3
DE67 3E11      LD  A,17      ;OVER TRACK
DE69 91      SUB  C      ;SECTER SU=17-SECTER

;
DE6A 47      DLOAD3:LD  B,A      ;B=SECTER SU
DE6B 3242DE      LD  (KAZU3),A      ;=KAZU3
DE6E 3A40DE      LD  A,(KAZU1)
DE71 90      SUB  B      ;A-B
DE72 3001      JR  NC,DLOAD4
DE74 AF      XOR  A
DE75 3240DE      DLOAD4:LD  (KAZU1),A

;
DE78 3E0E      LD  A,0EH      ;0EH=LOAD DATA
DE7A CDD8DE      CALL SENCOM      ;SEND COMMAND
DE7D 3A42DE      LD  A,(KAZU3)      ;SECTER SU
DE80 CDD8DE      CALL SENDAT      ;SEND DATA
DE83 3A3EDE      LD  A,(DRIVNO)      ;DRIVE NO.
DE86 CDD8DE      CALL SENDAT      ;SEND DATA
DE89 3A3FDE      LD  A,(TRACK)      ;TRACK NO.=P.TRACK*2+SURF.
DE8C CDD8DE      CALL SENDAT      ;SEND DATA
DE8F 3A3DDE      LD  A,(SECTER)      ;TOP SECTER
DE92 CDD8DE      CALL SENDAT      ;SEND DATA
DE95 3A3CDE      LD  A,(SC)      ;LOAD ADRESS HIGH  TRANS MEMORY TO
DE98 CDD8DE      CALL SENDAT      ;SEND DATA
DE9B AF      XOR  A      ;LOAD ADRESS LOW
DE9C CDD8DE      CALL SENDAT      ;SEND DATA

;
DE9F 3A40DE      LD  A,(KAZU1)      ;LEFT SECTER
DEA2 A7      AND  A
DEA3 281A      JR  Z,DLOADN      ;IF 0 THEN NEXT

;
DEA5 3A3CDE      LD  A,(SC)      ;MEMORY TOP MOVE
DEA8 47      LD  B,A
DEA9 3A42DE      LD  A,(KAZU3)
DEAC 80      ADD  A,B
DEAD 323CDE      LD  (SC),A
DEB0 3A3FDE      LD  A,(TRACK)      ;TRACK NEXT
DEB3 3C      INC  A
DEB4 3C      INC  A
DEB5 323FDE      LD  (TRACK),A
DEB8 3E01      LD  A,1
DEBA 323DDE      LD  (SECTER),A      ;SECTER=1
DEBD 189C      JR  DLOAD1

;
DEBF 3E15      DLOADN:LD  A,15H      ;15H=SEND DATA 2 BYTES
DEC1 CDD8DE      CALL SENCOM      ;SEND COMMAND

```



```

DEC4 3E50          LD  A,50H          ;MEMORY TOP HIGH
DEC6 C0DEDE        CALL SENDAT        ;SEND DATA
DEC9 AF            XOR  A              ;MEMORY TOP LOW
DECA C0DEDE        CALL SENDAT        ;SEND DATA
DECD 3A41DE        LD  A,(KAZU2)      ;SECTER SU
DED0 C0DEDE        CALL SENDAT        ;SEND DATA
DED3 AF            XOR  A              ;*100H
DED4 C0DEDE        CALL SENDAT        ;SEND DATA
DED7 C9            RET

;
; SEND COMMAND
;
DED8 F5            SENCOM: PUSH AF
DED9 3E0F          LD  A,0FH
DEDB D3FF          OUT  (0FFH),A
DEDD F1            POP  AF
DEDE C5            SENDAT: PUSH BC      ;SEND DATA
DEDF D5            PUSH DE
DEE0 F5            PUSH AF
DEE1 DBFE          SEND1: IN  A,(0FEH)
DEE3 E606          AND  6
DEE5 FE06          CP   6
DEE7 28F8          JR   Z,SEND1
DEE9 E602          AND  2
DEEB 28F4          JR   Z,SEND1
;
DEED 3E0E          LD  A,0EH
DEEF D3FF          OUT  (0FFH),A
DEF1 F1            POP  AF
DEF2 F5            PUSH AF
DEF3 D3FD          OUT  (0FDH),A
DEF5 3E09          LD  A,9
DEF7 D3FF          OUT  (0FFH),A
;
DEF9 DBFE          SEND4: IN  A,(0FEH)
DEFB E604          AND  4
DEFD 28FA          JR   Z,SEND4
;
DEFF 3E08          LD  A,8
DF01 D3FF          OUT  (0FFH),A
DF03 DBFE          SEND6: IN  A,(0FEH)
DF05 E604          AND  4
DF07 20FA          JR   NZ,SEND6
;
DF09 F1            SEND7: POP  AF
DF0A D1            POP  DE
DF0B C1            POP  BC
DF0C C9            RET
;
DF0D C5            RESDAT: PUSH BC      ;RESEEV DATA 2 BYTES OUT A,L
DF0E D5            PUSH DE
DF0F F5            PUSH AF
DF10 3E0B          LD  A,0BH          ;RFD BIT SET
DF12 D3FF          OUT  (0FFH),A
DF14 DBFE          RESV1: IN  A,(0FEH) ;DAV SET WAIT
DF16 E601          AND  1
DF18 28FA          JR   Z,RESV1
;
DF1A 3E0A          LD  A,0AH          ;RFD RESET
DF1C D3FF          OUT  (0FFH),A
DF1E F1            POP  AF
DF1F DBFC          IN  A,(0FCH)       ;RESCEV DATA A
DF21 F5            PUSH AF
DF22 3E0D          LD  A,0DH          ;DAC BIT SET
DF24 D3FF          OUT  (0FFH),A
DF26 DBFE          RESV2: IN  A,(0FEH) ;DAV RESET WAIT
DF28 E601          AND  1
DF2A 20FA          JR   NZ,RESV2
;
DF2C DBFC          IN  A,(0FCH)       ;RESCEV DATA L
DF2E 6F            LD  L,A
DF2F 3E0C          LD  A,0CH          ;DAC RESET
DF31 D3FF          OUT  (0FFH),A

```



```

DF33 18D4          JR    SEND7
      ;
      ;
DF35              END

```

## リスト2-4-1b 入力用プログラム

```

DE00 7E 32 3E DE 23 CD 43 DE 7E 23 66 6F 22 36 DE CD
DE10 56 DE 3A 41 DE 67 2E 00 22 38 DE ED 5B 36 DE CD
DE20 0D DF 12 13 7D 12 13 2A 38 DE 2B 2B 22 38 DE 7C
DE30 B5 C8 18 EB 00 00 00 00 00 00 00 00 00 00 00
DE40 00 00 00 7E 32 3F DE 23 7E 32 3D DE 23 7E 32 40
DE50 DE 32 41 DE 23 C9 3E 50 32 3C DE 3A 3D DE 4F 3A
DE60 40 DE 81 FE 12 38 03 3E 11 91 47 32 42 DE 3A 40
DE70 DE 90 30 01 AF 32 40 DE 3E 0E CD D8 DE 3A 42 DE
DE80 CD DE DE 3A 3E DE CD DE DE 3A 3F DE CD DE DE 3A
DE90 3D DE CD DE DE 3A 3C DE CD DE DE AF CD DE DE 3A
DEA0 40 DE A7 28 1A 3A 3C DE 47 3A 42 DE 80 32 3C DE
DEB0 3A 3F DE 3C 3C 32 3F DE 3E 01 32 3D DE 18 9C 3E
DEC0 15 CD D8 DE 3E 50 CD DE DE AF CD DE DE 3A 41 DE
DED0 CD DE DE AF CD DE DE C9 F5 3E 0F D3 FF F1 C5 D5
DEE0 F5 DB FE E6 06 FE 06 28 F8 E6 02 28 F4 3E 0E D3
DEF0 FF F1 F5 D3 FD 3E 09 D3 FF DB FE E6 04 28 FA 3E
DF00 08 D3 FF DB FE E6 04 20 FA F1 D1 C1 C9 C5 D5 F5
DF10 3E 0B D3 FF DB FE E6 01 28 FA 3E 0A D3 FF F1 DB
DF20 FC F5 3E 0D D3 FF DB FE E6 01 20 FA DB FC 6F 3E
DF30 0C D3 FF 18 D4 00 00 00 00 00 00 00 00 00 00
DF40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DF50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DF60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DF70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DF80 21 A0 DF CD 00 DE C9 00 00 00 00 00 00 00 00
DF90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DFA0 00 00 01 02 00 C0 00 00 00 00 00 00 00 00 00
DFB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DFC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DFD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DFE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DFF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

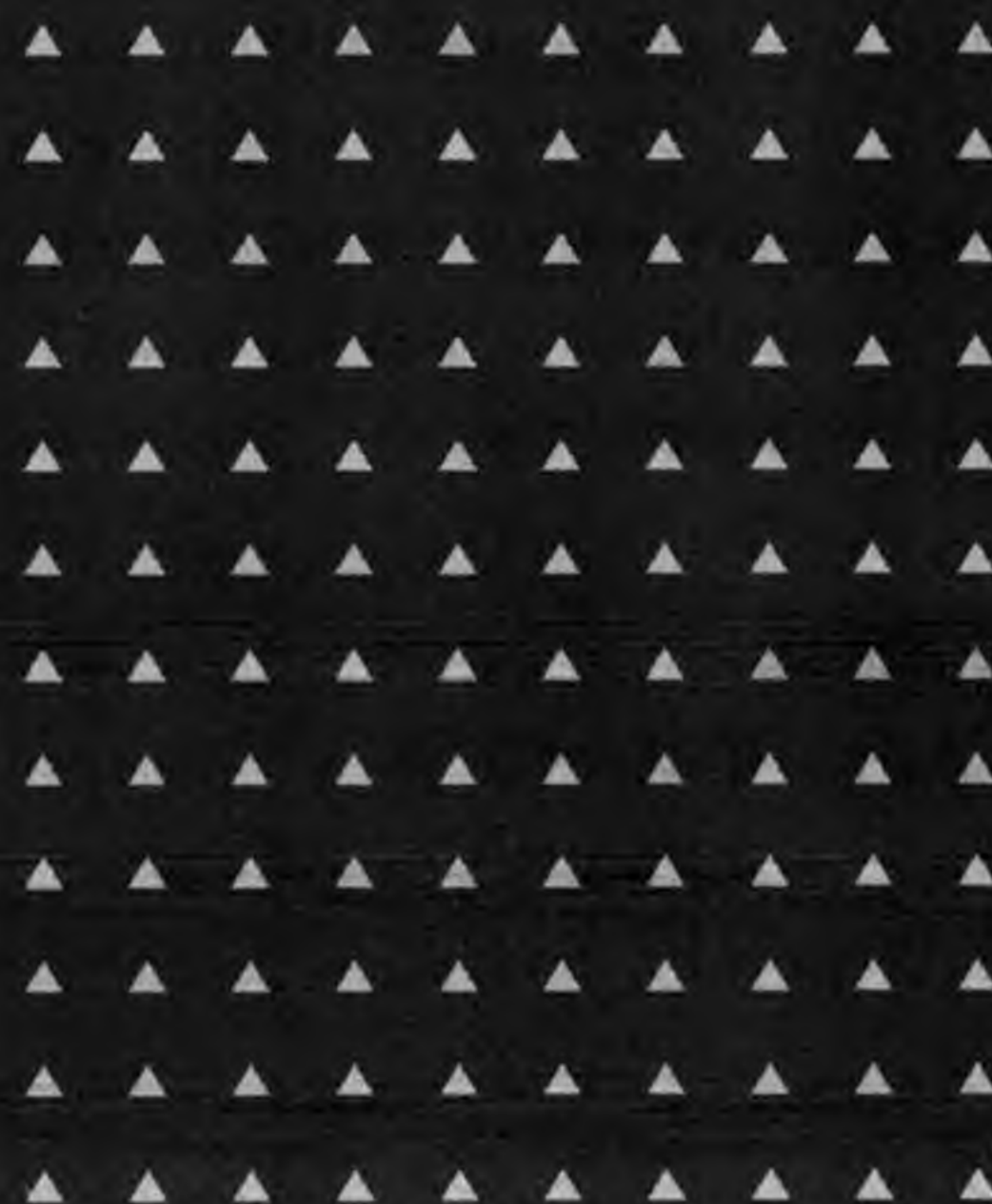
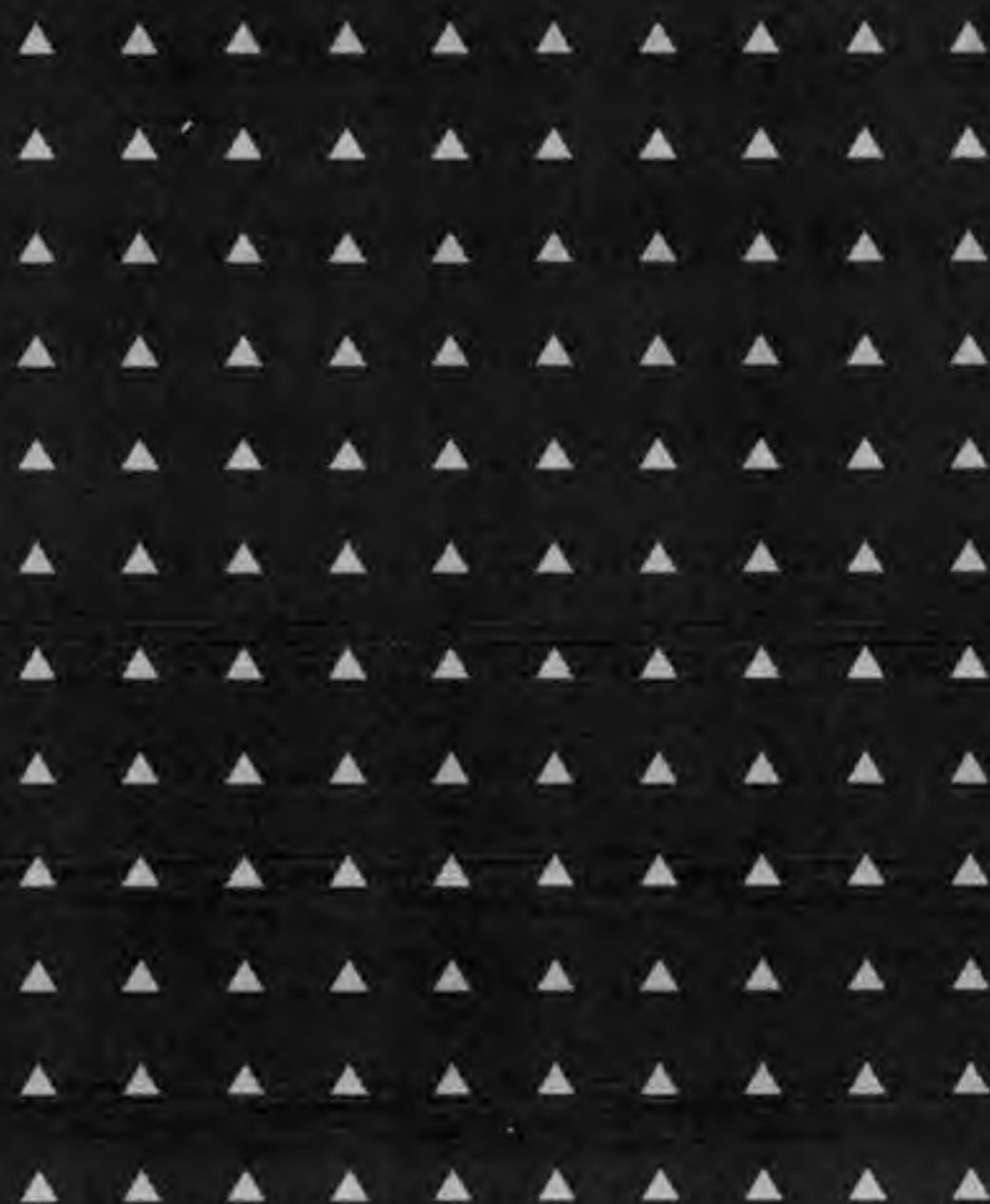
```



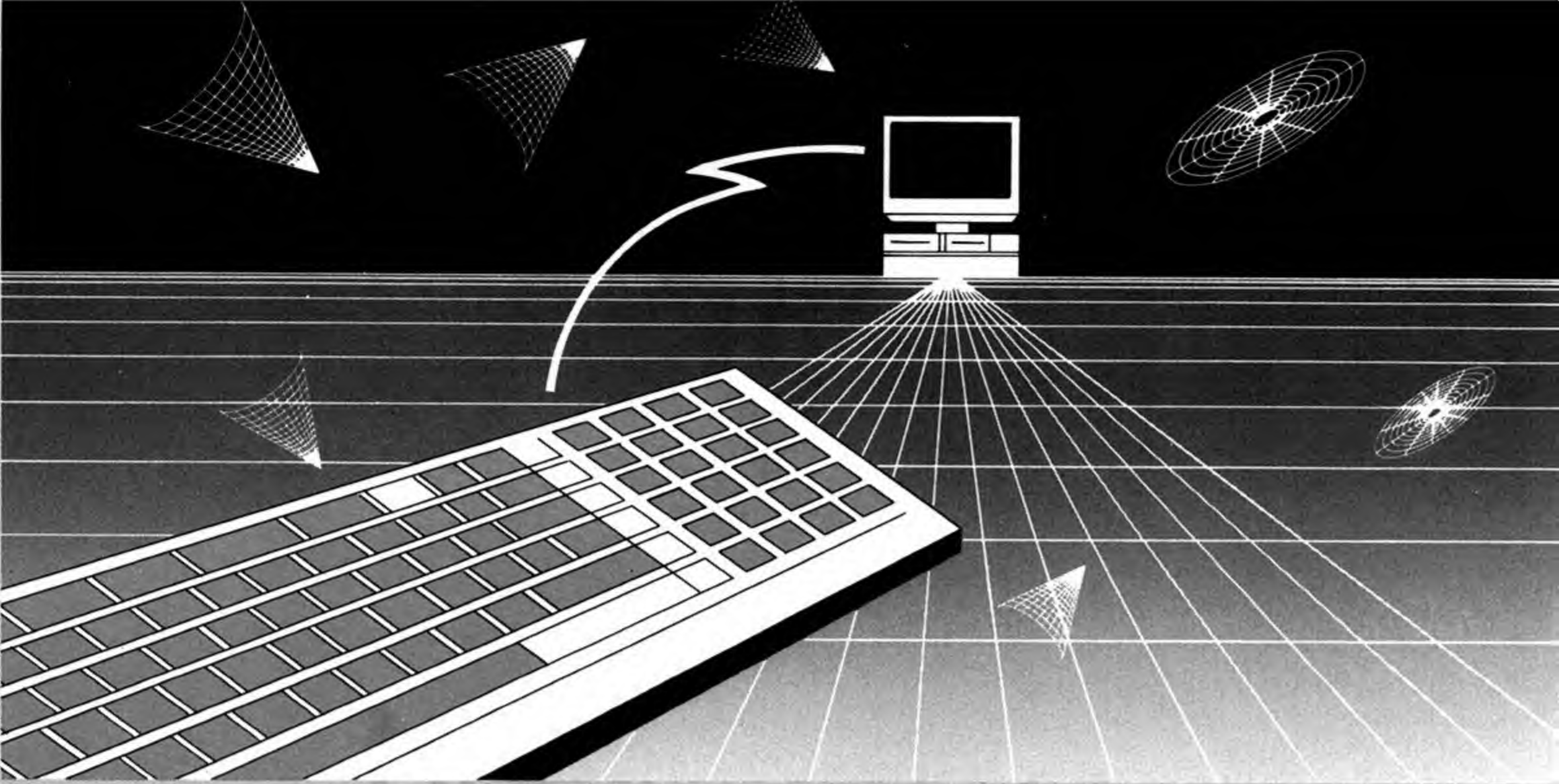
# 第 3 章

## BGMプログラミング

- 3-1 割り込みとは？
- 3-2 FM音源活用法
- 3-3 FM音源BGMルーチン







## 3-1 割り込みとは？

### 3-1-1

#### PC-88はマルチCPU？

みなさんは、BGMという言葉を知っているでしょう？ これは、Back Ground Musicの頭の文字をとったもので、何かをしているときに流す音楽のことです。

例えば、宿題をしながら歌謡曲を聞くとか、車を運転しながらロックを聞く、などの音楽のことです。

これはもういつもやっていることですよー。

これをパソコンにしてみると、

“あるプログラムを実行しながら、曲を演奏する”となります。プログラムを実行しながら、他のプログラムを実行する……。

ちょっと待ってください。我々のパソコンは、そんなマルチCPU的なことは、できましたっけ？

答えは、できます。それじゃーどーやるんだ！と、疑問に思った人、こーやるんです。

今CPUが、一つのプログラムを実行していたとしましょう。そのCPUに、定期的に他のプログラムを実行させたらどうでしょうか(第3-1-1図)。

こうすれば、何とかマルチCPU的な動作を、一つのCPUでできるのではないのでしょうか？

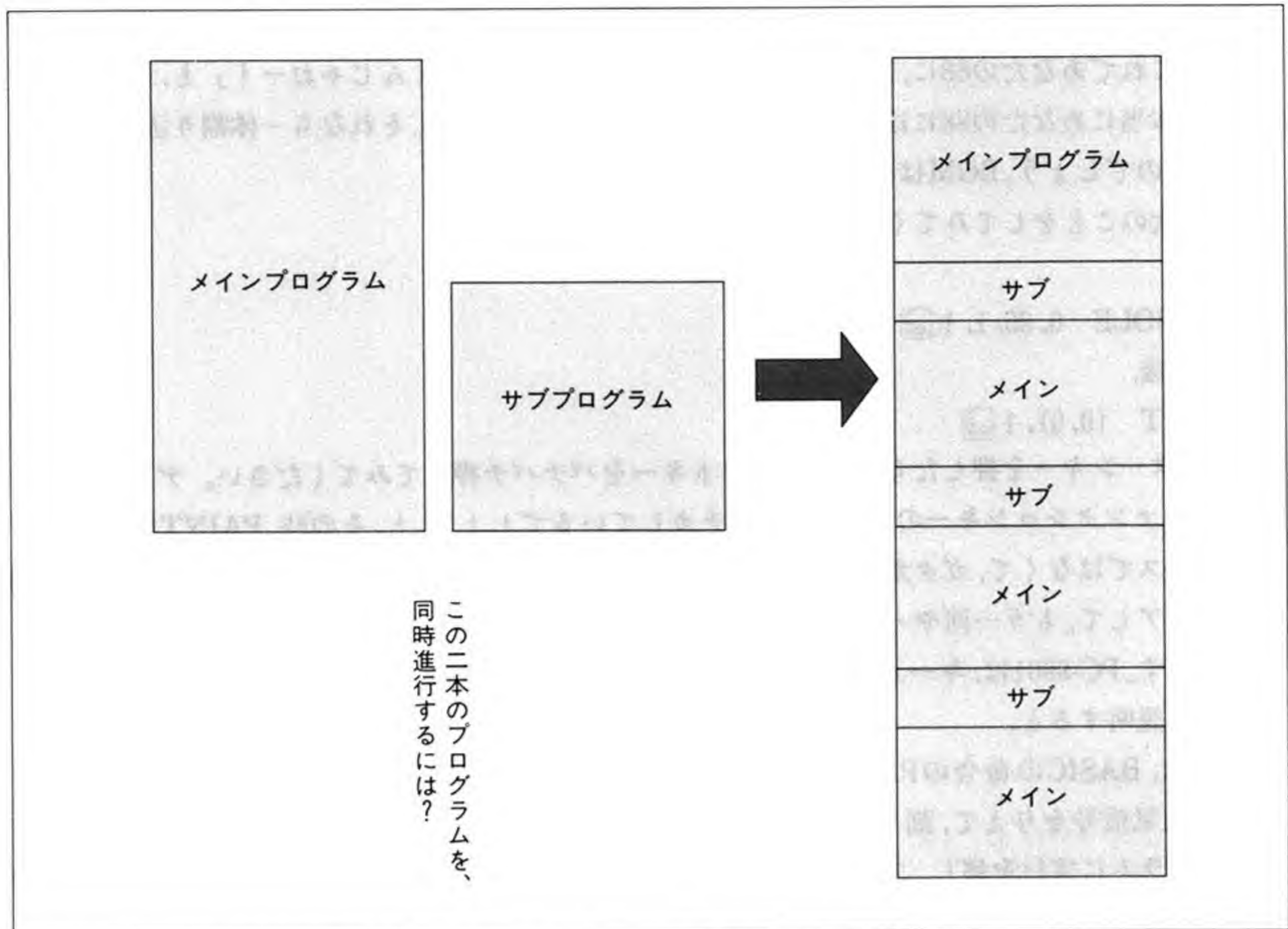
そして、CPUが定期的に実行する他のプログラムで音楽演奏をすれば、ほら、これでBGMつきパソコンのできあがりです。

### 3-1-2

#### 割り込みってなーに？

割り込みとは、私達の考えでは、





第3-1-1図 一つのCPUで、二つのプログラムを同時に実行するには？

“長い長い行列にならんでいると、前の方に二、三人若い男が割り込んだ”

とでもなるでしょう。パソコンの場合もほとんど同じで、

“長い長いプログラムを実行していると、CPUにおよびがかかり、他のプログラムを実行しなくてはならなくなった”

となります。これが、割り込みです。

人に例えてみると、本を読んでいる時に電話がかかってきたとか、食事をしているときに友達がやってきた。などです。

数多い、BGM付きのゲームは、この電話に出たり友達と話したりするときに、音楽演奏をしています。

### 3-1-3

## PC-8801に割り込みをかけるには？

さあ、BGMは、CPUに割り込みをかけて演奏することが分かりました。

それではどうやってあなたの88に割り込みをかけるのでしょうか？

まず、あなたの手元にPC-88シリーズのパソコンを持って来て下さい。次に、DISK BASICを立ち上げて、

“OK”




を表示させてください。


はい、これであなたの88に、割り込みがかかりました。「うそつくんじゃないー!」と、言わないで下さい。本当にあなたの88には、割り込みがかかっているんですよ。それなら一体割り込みで、何をしているのでしょうか。BGMは演奏していませんし。

では、次のことをしてみてください。

まず、

CONSOLE 0,25,1,1 

とした後、

PAINT (0,0),1 

で、リターンキーを押したら、すぐにシフトキーをバチバチ押してみてください。ディスプレイの下ファンクションキーの表示が、チカチカしているでしょう。と、その時、PAINTのスピードもスムーズではなくて、ガクガクしているのに気づきましたか? 分からない人は、グラフィック画面をクリアして、もう一回やってみましょう。

そうです。PC-8801は、キー入力のために、いつも割り込みがかかっているのです。

詳しく説明すると、

CPUは、BASICの命令のPAINT文を実行していました。ところが、CPU以外の他の回路が、CPUに電気信号を与えて、割り込みをかけます(第3-1-2図)。すると、CPUは指定されたアドレスのプログラムに実行を移し、ここでキーボードのキーが押されているかいないかを調べて、何も押されていないければ、そのままPAINTの仕事にもどります。

何かキーが押されていれば、そのキーのデータをバッファに入れてからPAINTの仕事にもどります。

もしも、押されていたキーがシフトキーで、ファンクション表示をしているなら、その内容をF・6からF・10のデータに取りかえて表示します。

さっきシフトキーを押して、PAINTのスピードが少し変化したのは、このファンクション表示の変更にかかる時間がかかってしまったからだったのです。

### 3-1-4

## みんなで割り込みを使おう!

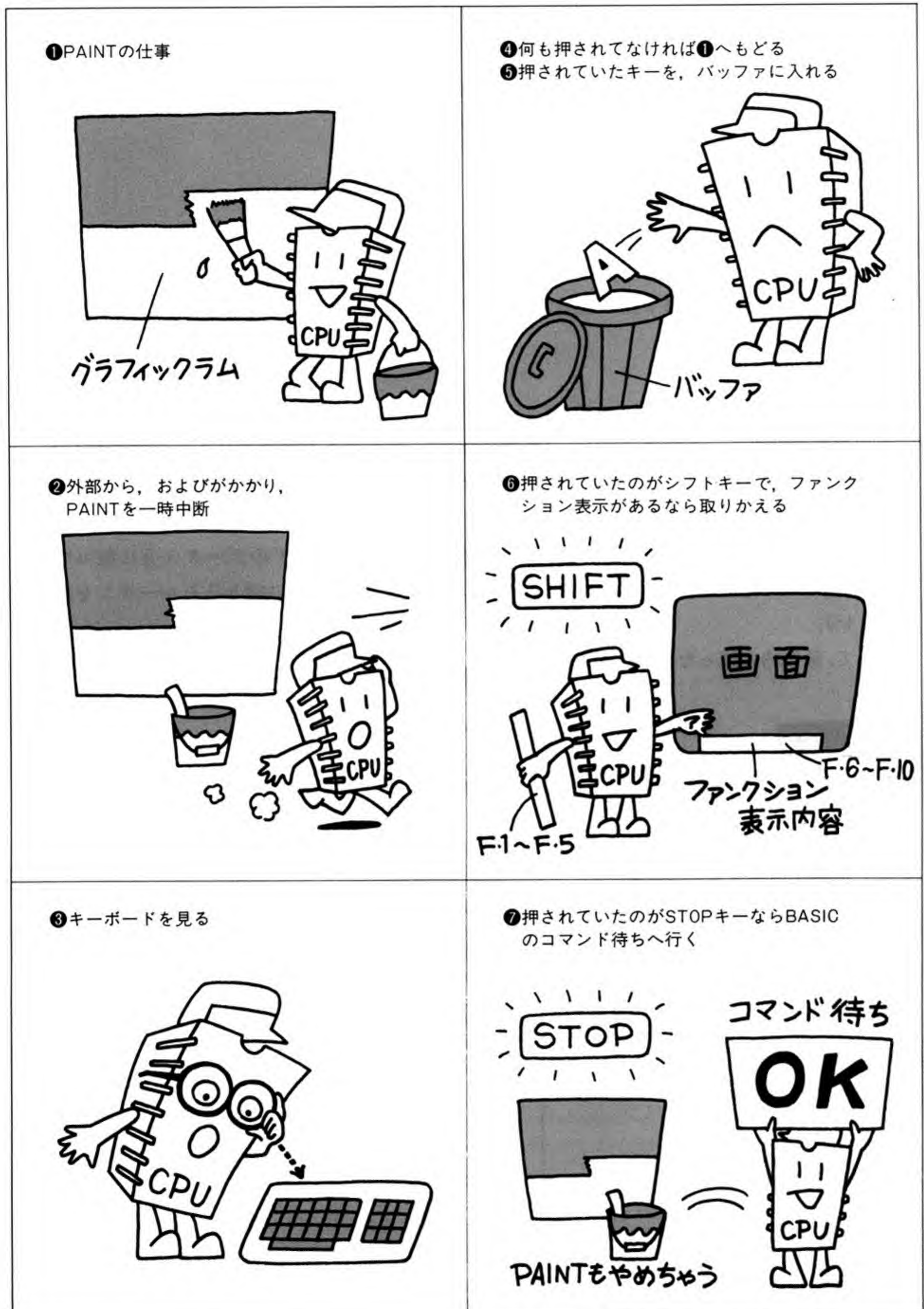
それでは、私達が自由に割り込みを使うには、どうすればよいのでしょうか?

N88-BASICでは、キーボード・RS-232C・拡張スロットの8インチフロッピーなどの機能を、割り込みで処理しています。もちろん、割り込みを使わなくてもこれらの機器を操作できます。

BGMを演奏するために割り込みを使うわけですが、CPUに割り込みをかけることができる機器は、

- ・本体内の回路
- ・画面表示のCRTC
- ・RS-232C
- ・拡張スロットバスの機器(8インチフロッピーなど)





第3-1-2図 絵で見る割り込み例



- ・FM音源のLSI

等があります。

しかし、定期的にCPUに割り込みをかけられるのは、どれでしょう。それは、

- ・本体内の回路
- ・画面表示のCRTC
- ・(FM音源)

です。なんでFM音源は、BGMを演奏するのに最も適しているのに、( ) がしてあるんだ？と、疑問に思うかもしれませんが、FM音源がCPUに割り込みをかけるのは、ちょっとやっかいで、難しいのです。そこで、一番手軽に使用できる割り込み発生機を紹介しましょう。

- ・本体内部の回路
- ・画面表示のCRTC

この二つです。

本体内部の回路というのは、CPUに、1秒間に600回も割り込みを発生させることができます。

画面表示のCRTCとは、ディスプレイに画面データ一枚分を送った後、CPUに割り込みをかけることができる機器です。具体的には、一秒間に60回割り込みをかけることができます。

しかし、この一秒間に60回の割り込みは、N88BASICがキーボードのデータ入力に使っているため、私達は使えません（本当は使えるが、一つの割り込みで、二つの割り込みルーチンを使うのは難しい）。

そして、最終的に残ったのは、一秒間に600回も割り込みを発生する、本体内部の回路です。

### 3-1-5

## PC-8801の割り込みに関するハード

さあ、それではこの600Hzのリアルタイム割り込みを使って、すこし遊んでみましょう。

リスト3-1を入力して、実行して見て下さい。

RUN 

の直後、OKの表示とともに、テキスト画面の一番上の左に、

“PC-8801”

という文字が現れたでしょう？

それでは、この文字を消してみして下さい。

どうです？ 全々消えないでしょう。CLRキーもダメなら、スペースキーもダメ。なぜでしょう。

リスト3-1 リアルタイム割り込み・サンプルプログラム

```
10 REM *** リアルタイム ワリコミ サンプル ***
20 CLEAR ,&HE4FF
30 AD=&HE500
40 READ A$:IF A$="END" THEN 80
50 POKE AD,VAL("&H"+A$)
60 AD=AD+1
70 GOTO 40
80 POKE &HF304,0
90 POKE &HF305,&HE5
100 OUT &HE6,7
110 END
120 DATA F3,C5,D5,E5,F5,3E,FF,D3,E4
130 DATA 21,1A,E5,11,C8,F3,01,07,00
140 DATA ED,B0,F1,E1,D1,C1,FB,C9,50
150 DATA 43,2D,38,38,30,31,END
```



それは、この“PC-8801”という文字は、さっきの割り込みを使用し、1秒間に600回もテキスト画面に書き込まれているのです。これを消すには、次のように打ち込んでください。

OUT &HE6,6

これで、リアルタイム割り込みはなくなって、文字も消せるようになりました。

このOUT命令の意味は後で説明します。

サンプルプログラムで分かったように、N88 BASICが動いているときも、私達が自由にリアルタイム割り込みを使えます。

さあ、いよいよCPUと、その周辺の割り込みに関するハードウェアと、I/Oポートを説明します。

第3-1-3図をみて下さい。PC-8801には、割り込みに関するハードがずいぶんありますね。

このうちで、各種の割り込みをコントロールしているのが、割り込みコントローラーと呼ばれるLSIで、I/Oポートアドレスは、&HE4です。

そしてもう一つ。今度は、I/Oアドレス&HE6にある、割り込み管理人です。

I/Oポート &HE4につながっている割り込みコントローラーとは、リアルタイム割り込みや、CRT割り込み、RS-232C割り込みなど、たくさんの割り込みの優先順位を決めるものです。

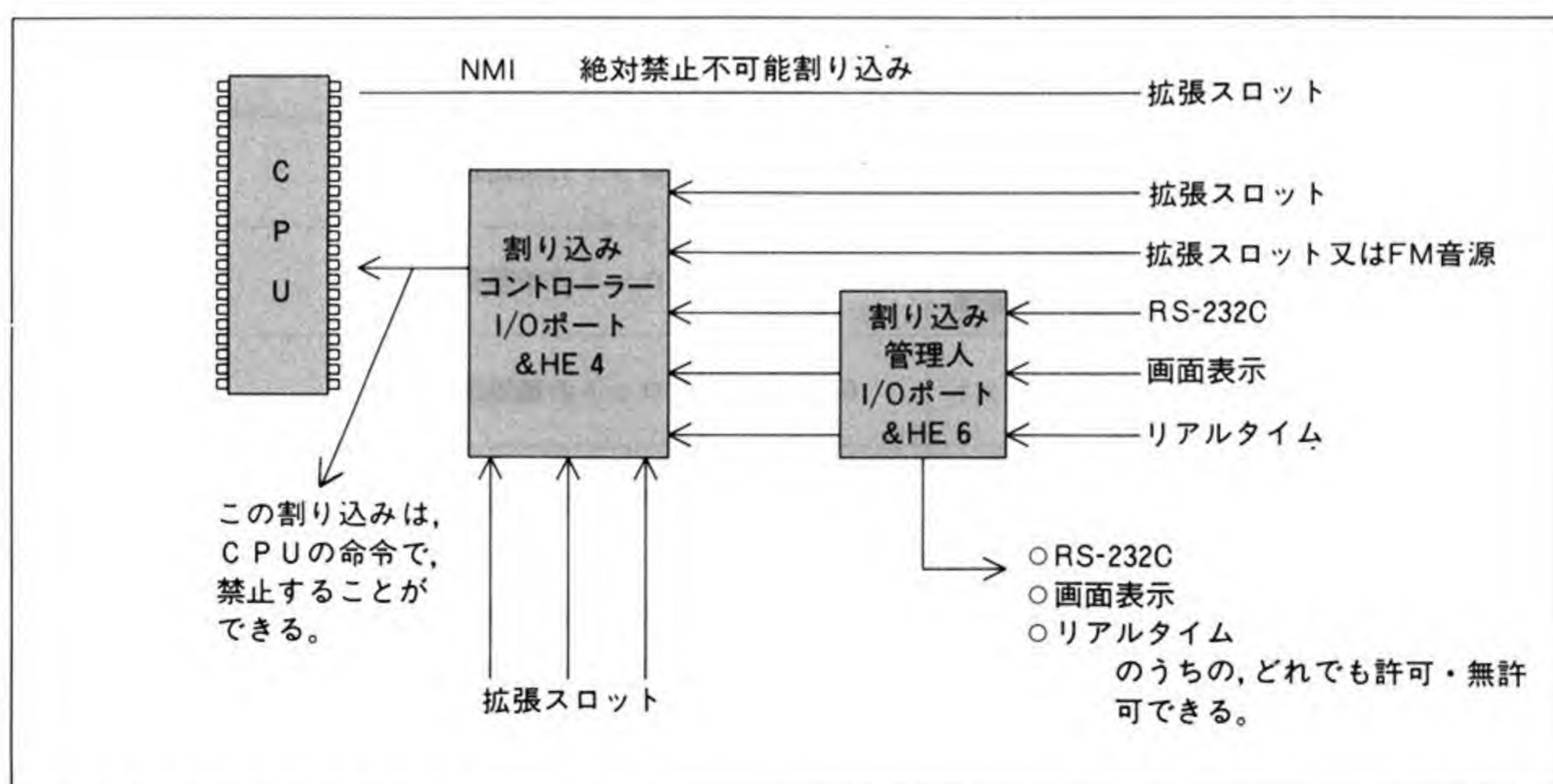
例えば、RS-232C割り込みと、リアルタイム割り込みが、同時に発生したとします。CPUには二つ同時には割り込みをかけられないので、二つのうちのどちらを先にCPUに与えるのかを決定します。

RS-232Cと、リアルタイム割り込みでは、RS-232Cのほうが優先されるので、割り込みコントローラーは、CPUにRS-232Cの割り込みをかけます。

このあたりが、とても難しいところです。

本章の最後に、割り込みコントローラーと、割り込み管理人の説明がしてありますので、詳しく知りたい人は、そちらを見て下さい。

さあ、それではいよいよ割り込みの使い方を説明いたします。



第3-1-3図 PC-8801の割り込みハード



## 3-1-6

## 割り込みテーブルと、Iレジスタ

いきなり難しいタイトルが出てきましたが、心配しなくても大丈夫です。

さっきの第3-1-3図を見て下さい。CPUに割り込みをかけられる機器は、最大で9個までです。このうちで、PC-8801に標準実装されているのは、たったの四つです。残りの五つは、すべて拡張スロットのためのものです。

第3-1-4図をみて下さい。これが割り込みテーブルと呼ばれているものです。

このテーブルは、N88BASICではメモリの&HF300からはいっています。割り込みがかかるとCPUは、まずこのテーブルを見ます。そして、割り込みをかけた周辺機器によってアドレスを読み込み、そのアドレスをコールします。

例をあげましょう。

今CPUが、リストを表示していました。そこへ、リアルタイム割り込みがかかりました。するとCPUは、&HF304と&HF305のアドレスの内容を持ってきます。普通の場合、F304には&H0Eが、F305には&HE8が書かれています。次にCPUは、この&H0Eと&HE8を逆にとりかえたアドレス、つまり&HE80E番地をコールします。

コールというのは、ジャンプとは違って、ジャンプするのがGOTOならば、コールはGOSUBです。つまりサブルーチンの最後に、リターン命令があればサブルーチンへ来る前のプログラムにも

優先順位	アドレス (16進)	メモリの内容	割り込み名	割り込みレベル	説明
高 ↑      ↓ 低	F30 <sub>0</sub> 1	EA E7	RS-232C	1	RS-232Cの割り込みがかかると、CPUは&HE7EAをコールする
	F30 <sub>2</sub> 3	08 E8	画面表示	2	画面表示割り込みがかかると、CPUは&HE808をコールする（キー入力している）
	F30 <sub>4</sub> 5	0E E8	リアルタイム	3	リアルタイム割り込みがかかると、CPUは&HE80Eをコールする
	F30 <sub>6</sub> 7	14 E8	拡張スロット	4	拡張スロットの機器用
	F30 <sub>8</sub> 9	14 E8	拡張スロット 又はFM音源	5	拡張スロットの機器用
	F30 <sub>A</sub> B	14 E8	拡張スロット	6	拡張スロットの機器用
	F30 <sub>C</sub> D	1A E8	8インチディスク	7	8インチディスク用
	F30 <sub>E</sub> F	20 E8	8インチディスク	8	8インチディスク用

この割り込みテーブルは、モニタで見ることができます。

この表の見かたは、例えば、リアルタイム割り込みがかかると、CPUは&HF304と、&HF305の内容を見ます。そして、その内容の示すアドレスをCALLします→&HE80EをCALL。

第3-1-4図 N88-BASICの割り込みテーブル



どってくるのです。

ところで第3-1-3図では、CPUに割り込みをかけられる機器は9個までつけられるのに、どうして割り込みテーブルは、8個までしかないのでしょうか。

それは第3-1-3図の絶対禁止不可能割り込みに関係があります。この絶対禁止不可能割り込みは、NMIと呼ばれていて、CPUがどんな状態であろうとこの割り込みがかかると、メモリの&H0066番地をコールします。

他の割り込みは、CPUの命令でCPUに割り込みをかけてもCPUはしらん顔してプログラムをつづけます。この命令をDI(Disable Interrupt)といいます。

逆に、CPUに割り込みをかけて、割り込みルーチンへコールするようにするCPUの命令は、EI(Enable Interrupt)といいます。

割り込みテーブルを使用する割り込みは、すべてこの命令に左右されます。つまり、CPUがDIの状態の時に、いくらRS-232C割り込みなどをかけてもダメです。

これに対してNMIの方は、その名のとおり絶対禁止不可能ですから、CPUにこの割り込みがかかると必ず&H0066番地をコールします。

ところで、CPUに何か(NMIは除く)の割り込みがかかると、&HF300番地からの割り込みテーブルを使用しますが、このテーブルのアドレスは移動できないのでしょうか？ PC-8801では、N88モードの時は、割り込みテーブルは&HF300からですが、Nモードのときは&H8000からです。

ということは、何かをすればテーブルの場所が移動するはずで、そこで登場するのが、Iレジスタです。

この「Iレジスタ」とは、割り込みテーブルのあるアドレスの上位8ビットを記憶しているものです。

簡単にいうと、割り込みテーブルが&HF300にあるなら、Iレジスタというのは&HF3が、テーブルが&H8000なら、Iレジスタの内容は&H80だということです。

このIレジスタは、CPUがCPUの中で記憶してあります。したがって、このIレジスタの内容を変更すれば、自動的に割り込みテーブルの上位アドレスも変わります。しかし、下位8ビットは、必ず00ですので、テーブルを&H××33番地とか、&H××F0とかはできません。

また、第3-1-4図のように、割り込みテーブルの中では、順番は変えられません。

## 3-1-7

### まとめ

#### ●割り込みコントローラー I/O &HE4

○このコントローラーで、CPUに割り込みをかけることのできる周辺機器を限定します。

#### ●割り込み管理人 I/O &HE6

○このポートで、リアルタイム、画面表示、RS-232C、の三つの割り込みの、OK、NOを区別します。

#### ●CPUの命令 DIとEI



- CPUがDI状態のときは、NMI割り込み以外の割り込みはCPUは全く受け付けません。  
○CPUがEI状態のときは、いつでも受け付けます。

## まとめ1●割り込みコントローラー

I/Oポート &HE4 OUTのみで、INは意味なし

0	すべて禁止
1	RS-232C割り込みまで許可
2	画面表示割り込みまで許可
3	リアルタイム割り込みまで許可
4	拡張スロットINT4まで許可
5	拡張スロットINT3まで許可
6	拡張スロットINT2まで許可
7	拡張スロットFDINT1まで許可
8	拡張スロットFDINT2まで許可

I/Oポート &HE4 に出力した値までの割り込みを許可します。

つまり、

OUT &HE4, 3

とすると、

- RS-232C
- 画面表示
- リアルタイム

の三つの割り込みが許可されます。

- なお、OUT &HE4, 0を実行すると、いくら周辺機器が割り込みを発生しても、このコントローラーがそれをカットしてしまうので、CPUにはつたわりません。
- 割り込みの優先順位は、表の番号が少ないほど高くなっています。
- 割り込みルーチンでは、このコントローラーに必ず適当な値をOUTして下さい。そうしないと、CPUに割り込みがつたわらなくなります。

## まとめ2●割り込み管理人

I/Oポート &HE6 OUTのみでINは意味無し。

					RS-232C BIT2	画面表示 BIT1	リアルタイム BIT0
--	--	--	--	--	-----------------	--------------	----------------

○斜線部のビットは意味を持たない。


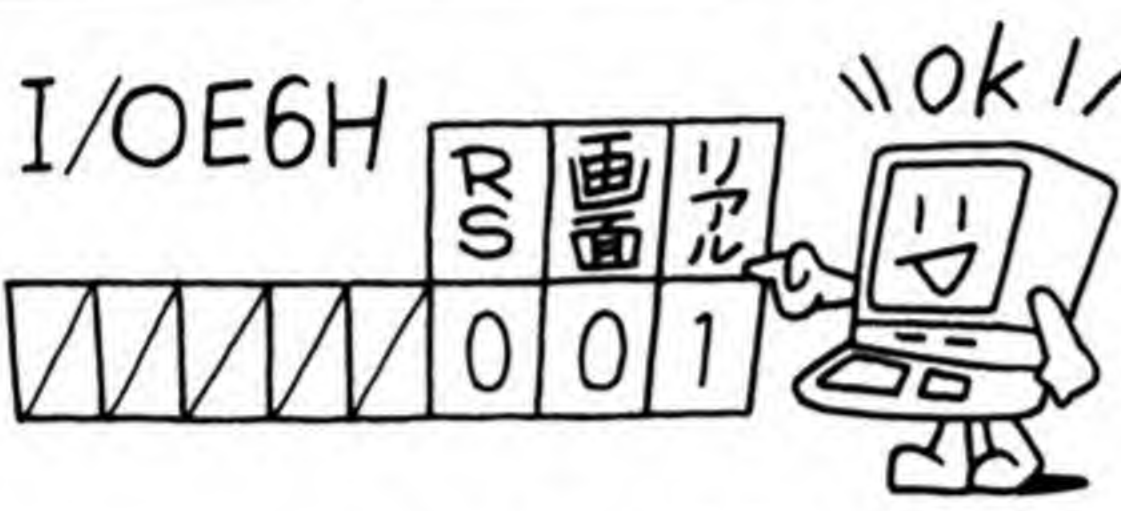
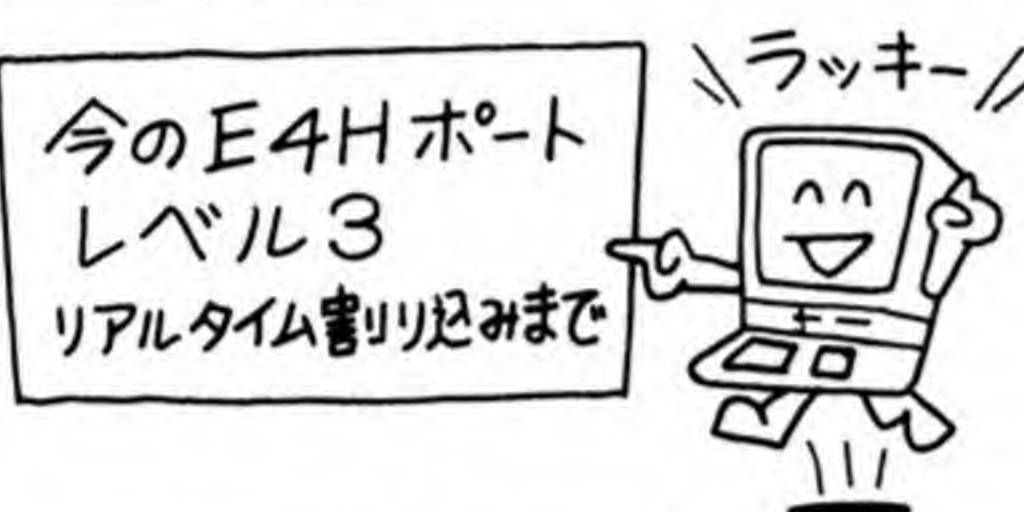



- RS-232C BIT2      1 ……RS-232C割り込み許可  
                          0 ……RS-232C割り込みダメ
- 画面表示 BIT1      1 ……画面表示割り込み許可  
                          0 ……画面表示割り込みダメ
- リアルタイム BIT0   1 ……リアルタイム割り込み許可  
                          0 ……リアルタイム割り込みダメ

OUT &HE6 に出力する値の下3ビットが意味を持つ。つまり、画面表示割り込みのみを許可したいのなら、

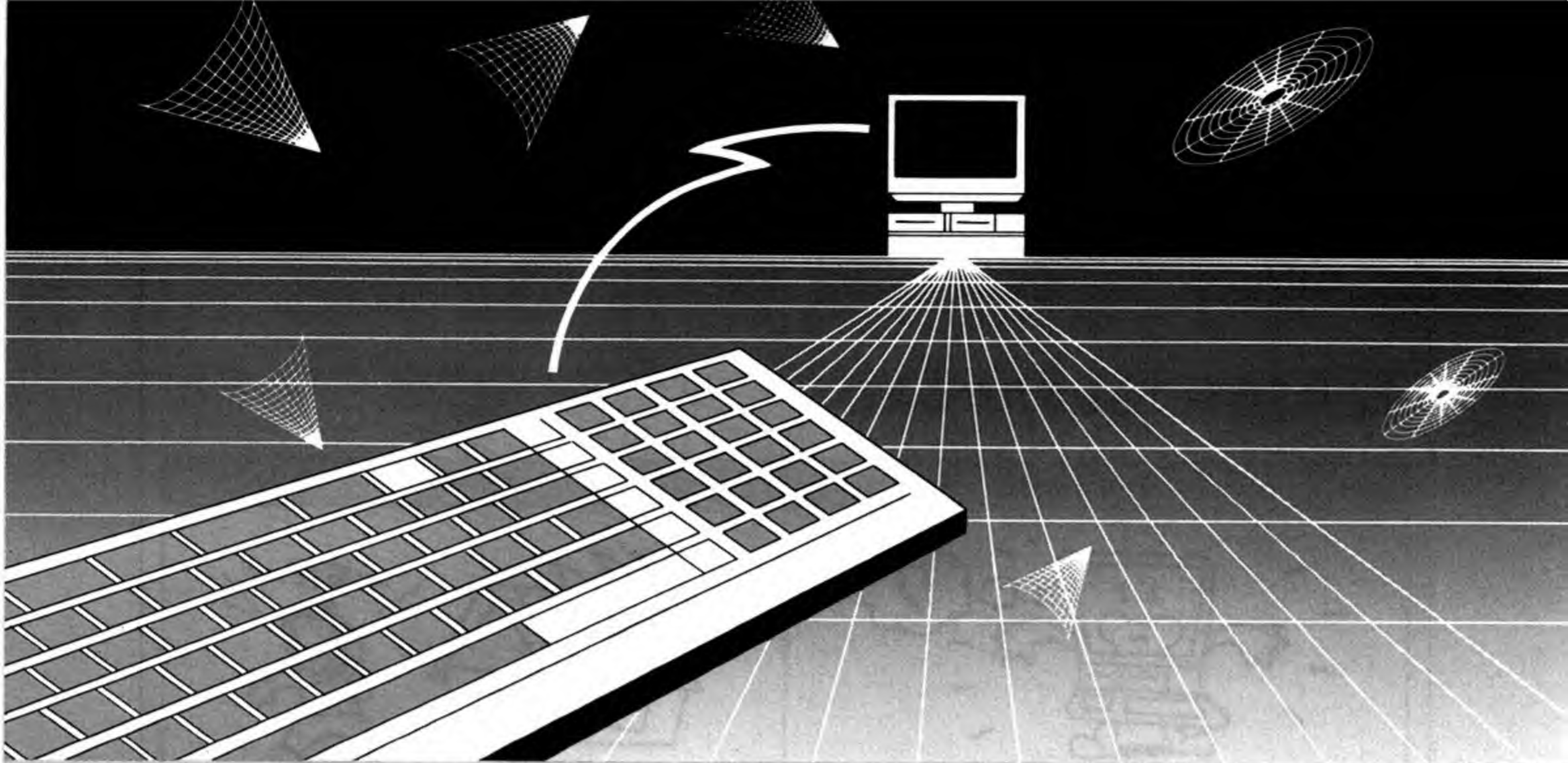


OUT &HE6, 2  
と、すばしい。

### まとめ3●割り込みがかかるまで

コンピュータの場合	人間の場合
 <p>リアルタイム割り込み発生</p>	<p>緊急事態発生!</p>
 <p>&amp;HE6ポートの通過!</p>	<p>橋の体重制限通過!</p>
 <p>&amp;HE4ポートの通過!</p>	<p>門の年齢制限通過!</p>
 <p>CPUは今EI状態</p>	 <p>消防所は今仕事中</p>
 <p>CPUは、割り込みルーチンへ</p>	<p>出動!</p>





## 3-2 FM音源活用法

### 3-2-1

#### いよいよFM音源!

FM音源は、NECのパソコンでは、SR以降の8ビット機に、必ず搭載されています。また、それ以前の機種にも、FM音源ボードとして発売されています。

FM音源のLSIは、今ではPSGに代わって、コンピュータ・ミュージックにかかすことのできないものになりました。

それに伴って、FM音源に関する本も数多く出まわっています。

ここでは、FM音源について次の説明をいたします。

- FM音源の使い方
- マシン語による音色設定
- メモリ・データにした音色データ
- PLAY文を使わずに音を出す

以上です。

そして、3-3章では割り込みを使用したBGMルーチンと、そのルーチンをフルに発揮するユーティリティを説明します。

さあそれでは話を進めましょう。

### 3-2-2

#### FM音源の使い方

FM音源の使い方といっても、もちろんBASICのPLAY文や、VOICE文で使うものではありません。ここではマシン語で使います。そんな嫌な顔をしないで下さい。簡単なんです。本当に。



## ※ WRITE DATA

## ADDRESS

21H	TEST			
24H	TIMER-A			
25H				TIMER A
26H	TIMER-B			
27H	MODE	RESET B A	ENABLE B A	LOAD B A
28H	SLOT			CH
2DH				
2EH				
2FH				
30H		DT	MULTI	
3EH				
40H		TL		
4EH				
50H	KS		AR	
5EH				
60H			DR	
6EH				
70H			SR	
7EH				
80H	SL		RR	
8EH				
90H			SSG-EG	
9EH				
A0H	F-Num. 1			
A1H				
A2H				
A4H		BLOCK	F-Num. 2	
A5H				
A6H				
A8H	3CH * F-Num. 1			
A9H				
AAH				
ACH		3CH * BLOCK	3CH * F-Num. 2	
ADH				
AEH				
B0H		FB	CONNECT	
B1H				
B2H				

## CONNECT

## LSIの TEST-DATA

TIMER-Aの上位8bits

TIMER-Aの下位2bits

TIMER-Bの DATA

TIMER-A/BのControlおよび3CHのMode  
key-on/off

プリ スケーラをSet

1/3, 1/6分周の選択

分周器を1/2分周にセット

Detune/Multiple

(33H, 37H, 3BHの Addressは無し)

Total Level

(43H, 47H, 4BHの Addressは無し)

key Scale / Attack Rate

(53H, 57H, 5BHの Addressは無し)

Decay Rate

(63H, 67H, 6BHの Addressは無し)

Sustain Rate

(73H, 77H, 7BHの Addressは無し)

Sustain Level / Release Rate

(83H, 87H, 8BHの Addressは無し)

SSG-Type Envelop Control

(93H, 97H, 9BHの Addressは無し)

F-Numbers/BLOCK

3CH-3slot

F-Numbers/BLOCK

Self-Feedback/Connection

第3-2-1図 OPNアドレスマップ1



マシン語の命令なんて分からなくても、リストのとおりに入力して、説明のとおりに行えばいいんです。

PC-8801mk IISR/FR/MR/FH/MHには、YAMAHAのFM音源OPNが入っています。

このLSIも割り込みコントローラーの時と同じように、I/Oポートにつながっています。I/Oポートアドレスは、&H44と&H45です。すなわち、FM音源は、一つのLSIで二つのI/Oポートを使っているのです。なお、88やmk II・80mk II用のFM音源ボード、PC-8801-11のI/Oポートアドレスは、&HA8と&HA9です。

一つのLSIで、二つのI/Oポートを使っているのはどうしてでしょう。それはこういうことです。

第3-2-1図を見てください。これはFM音源OPNのレジスタと呼ばれているもので、CPUのそれと同じようにOPNの中の記憶場所です。FM音源にデータを送るには、まずレジスタを指定し、そのあとにデータを送ります。

つまり、88のI/Oポート&H44でOPNのレジスタを指定して、&H45でデータを出力します。

今後のプログラムでも、この&H44と&H45ポートが頻繁に登場します。そしてさっきも書いたようにFM音源ボードを使用している人は、サンプルプログラムや、マシン語プログラムの中の44Hや45Hを、それぞれA8HとA9Hにすれば動作します。

### 3-2-3 マシン語による音出し

ここでは本当は、マシン語による音色設定を説明する予定だったのですが、その前にちょっと次のことを試してみてください。

まず、先程の第3-2-1図を見てください。この図の中には、20H～B2Hまでの数で示されているレジスタがあります。この中で、レジスタ番号28HのKey on / offと書いてあるところを見てください。知っている人も多いと思いますが、FM音源はこのレジスタを操作して音を出すのです。

Key on / offの意味は、ピアノの鍵盤を押した時がonで、離した時がoffです。つまり、このレジスタをonにすると、そのチャンネルの音色が出ます。次に音を出すときは、必ず一度offにしないと、音がでません。

さあ、それではこのレジスタをいじくって遊んでみましょう。リスト3-2-1を入力して下さい。そして、

#### リスト3-2-1 音出しサンプルプログラム

```

10 REM *** OPN Sample Key on / off ***
20 OUT &H44,&HA4:OUT &H45,&H1A
30 OUT &H44,&HA0:OUT &H45,&H69
40 OUT &H44,&H28          ' OPN のレジスタに 28H
50 OUT &H45,&HF0          ' チャンネル 1 を ON
60 FOR I=1 TO 1000:NEXT   ' Timer
70 OUT &H44,&H28
80 OUT &H45,&H0          ' チャンネル 1 を OFF
90 END

```



NEW CMD

とした後、何か自分の好きな音色を設定して下さい。私なら、

CMD PLAY "@13"

となるでしょう。

さあ、それではプログラムを

RUN

して下さい。

「ドーン」と音が出ましたか？ えっ出ないって？ ボリュームは大きくなっていますか？ FMボードの人は、I/Oアドレスを確かめましたか？

音が出たのはあたりまえのように感じているでしょうが、あなたはPLAY文の"C"や"D"を使いましたか？ 使っていないでしょう。このサンプルプログラムの中にもPLAY文はありません。

そうです。あなたは今、OUT命令だけで音を出したのです。もう一度RUNしますか？

BASICのOUT命令はすぐにマシン語になります。なぜならば、マシン語の命令の中にもBASICと同じOUT命令があるからです。

しかし、PLAY命令はどうでしょう。これはマシン語の命令の中にはありません。もしもマシン語でPLAY命令と同じことをやろうと思ったらとても大変でしょう。だからこれから書くサンプルプログラムも、なるべくPLAY文は使用せず、OUT命令で書くようにします。BASICプログラムは難しくなりますが、すぐにマシン語に変換できるプログラムになります。

### 3-2-4

#### マシン語による音色設定

さきほどのサンプルプログラムでは、OPNの

Key on / off

のみをOUT命令で実行しましたが、今度はOUT命令だけで、音色設定をしてみましょう。

リスト3-2-2がそれです。リストは、BASICとマシン語、両方とも入力して下さい。マシン語は、

CLEAR ,&HBFFF

を実行してから入力しましょう。

実はこのマシン語リストは、単なるデータで、マシン語プログラムではありません。そうです、このデータが、FM音源の音色データです。

BASICのプログラムが、メモリの音色データを読み込んで、OPNのレジスタに出力しています。

さあそれでは、プログラムをRUNしてみましょう（RUNする前に、マシン語データも読み込んでおいて下さい）。

RUNすると、

address=?

と、聞いてくるので、C000H以後の40Hの倍数、つまりC000Hとか、C280Hなどで答えて下さい。

ここでは初めて使うので、とりあえず

C000



と入力して下さい(&Hはいりません)。

ずらずらと文字が出て、一番最後にパープシコードの音で、

ドー

が聞こえたでしょう？

さあ、音色データはまだまだあります。

もう一度RUNして、今度は

address=?

には、

C340

と入力してみましょう。

しばらくして、ピアノの音が聞こえたでしょう。

さあ、やっとPLAY文を使わずに音色設定ができました。あとは音程の設定と音の長ささえコントロールできればよいのですが、先へ急ぐのはやめて、今のサンプルプログラムの説明をいたしましょう。

第3-2-2図、第3-2-3図を見て下さい。このサンプルプログラムは、これまたすぐにマシン語に変換できるプログラムです。このプログラムをRUNしたときのアドレス入力、音色データの入っているメモリのアドレスですが、このメモリの音色データはPC88のPLAY文の

"@0"

から、

"@61"

#### リスト3-2-2a 音色設定プログラム(BASIC部)

```

10 INPUT "address=";AD$
20 AD=VAL("&H"+AD$)
30 AF=AD+1
40 FOR RE=&H30 TO &HA0 STEP 4
50     DA=PEEK(AF):AF=AF+1
60     OUT &H44,RE
70     OUT &H45,DA
80     PRINT "OPN レジスタ ";HEX$(RE);"H"    out data ";HEX$(DA);"H"
90 NEXT
100 OUT &H44,&HB0                                'Connection
110 OUT &H45,PEEK(AD)
120 OUT &H44,&H2D                                'OPN INIT.
130 OUT &H44,&HA4                                'オンテイ セッテイ
140 OUT &H45,&H1A
150 OUT &H44,&HA0                                'オンテイ セッテイ
160 OUT &H45,&H69
170 '
180 '
190 OUT &H44,&H28:OUT &H45,&HF0                    'Key on
200 FOR I=1 TO 1000:NEXT
210 OUT &H44,&H28:OUT &H45,&H0                    'Key off
220 END

```



## リスト3-2-2b 音色設定プログラム(マシン語データ部)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C000:	3A	0C	01	1F	73	20	1E	39	00	1F	1F	DF	9F	0C	0C	02	:26
C010:	05	04	04	04	07	1A	06	F6	27	15	00	01	02	00	00	00	:6D
C020:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C030:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C040:	3A	01	01	2A	11	1A	36	1E	00	8E	4F	59	52	0A	0B	0D	:8F
C050:	03	00	00	03	00	15	25	FF	28	15	00	01	02	00	00	00	:7F
C060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C070:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C080:	3A	12	02	1A	12	1B	33	18	00	5C	5C	53	4D	07	07	09	:4F
C090:	04	00	00	00	00	13	13	32	07	15	00	01	02	00	00	00	:7B
C0A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C0B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C0C0:	3A	31	64	0C	01	21	31	47	00	9C	9C	DB	DA	04	04	09	:73
C0D0:	03	03	03	01	00	17	06	02	A5	15	00	01	02	00	00	00	:E6
C0E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C0F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	F7	57	6F	77	9E	CF	FC	DF	FB	F9	66	6A	20	21	22	21	:C4
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C100:	20	36	30	35	31	1C	16	3A	00	DF	9F	DF	9F	07	09	06	:6A
C110:	06	07	06	06	08	29	19	19	F9	15	00	01	02	00	00	00	:8D
C120:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C130:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C140:	3D	35	52	78	31	27	1B	11	00	DF	1F	1F	1F	12	0F	04	:21
C150:	0F	00	00	00	00	2F	0F	0F	0F	15	15	15	00	01	02	00	:AD
C160:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C170:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C180:	04	58	60	54	10	1F	1F	1B	00	94	92	90	92	02	02	02	:C7
C190:	02	00	00	00	00	07	05	07	06	15	15	00	01	02	00	00	:48
C1A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C1B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C1C0:	3B	56	33	06	03	28	26	33	00	DF	D0	54	8F	09	0B	07	:FB
C1D0:	04	03	00	00	00	E5	25	F5	08	15	00	01	02	00	00	00	:26
C1E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C1F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	B7	23	1B	0D	7D	CE	C8	BD	16	85	4A	F9	E4	27	27	13	:F5
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C200:	3A	31	32	39	34	1F	32	28	00	D9	56	DC	54	0B	00	0C	:F9
C210:	06	00	00	0C	00	13	1B	5B	0B	15	00	01	02	00	00	00	:BE
C220:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C230:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C240:	3B	01	04	02	01	27	33	24	00	5F	5F	13	53	00	0F	0B	:FF
C250:	0F	00	00	00	00	00	09	4B	09	15	00	01	02	00	00	00	:84
C260:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C270:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C280:	24	39	05	76	01	1E	41	19	00	5F	9E	DB	9E	10	0C	07	:EA
C290:	05	00	0B	0A	0A	BA	F6	85	F5	15	15	00	01	02	00	00	:7B
C2A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C2B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C2C0:	3A	0C	01	1F	73	20	1E	39	00	1F	1F	DF	9F	0C	0C	02	:26
C2D0:	05	04	04	04	07	1A	06	F6	27	15	00	01	02	00	00	00	:6D
C2E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C2F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	F2	7B	4B	EA	BA	6B	E4	BF	30	0A	87	AC	EB	29	27	20	:32
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C300:	1C	33	35	21	33	16	19	0F	00	1D	CA	1D	5E	04	04	04	:84



```

C310: 04 01 04 03 03 20 30 03 03 15 15 00 01 02 00 00 :92
C320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C340: 3C 71 23 11 61 20 1C 0A 00 DF DF 99 99 00 00 07 :7F
C350: 08 08 08 07 06 40 40 37 37 15 15 00 01 02 00 00 :40
C360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C380: 07 3F 5F 5F 3F 00 00 00 00 1E 0A 14 02 10 14 14 :B9
C390: 14 04 14 14 0E A7 07 07 07 15 15 15 15 00 01 02 :61
C3A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C3B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C3C0: 3E 0F 00 01 03 25 14 2A 00 44 44 44 4A 0A 0C 0A :EA
C3D0: 0C 02 02 02 02 04 03 04 03 15 15 15 00 01 02 00 :64
C3E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C3F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----
sum : C9 01 D9 B2 EF 66 C3 88 44 B2 4B 38 5A 23 27 2B :3D

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: 36 3F 55 53 3F 0F 19 00 0F 02 0E 04 0A 02 0E 08 :C9
C410: 04 02 05 03 0A 07 07 07 17 15 15 15 00 01 02 00 :86
C420: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C430: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C440: 3A 00 30 30 34 0E 06 14 00 4C 9D 50 89 05 06 04 :C7
C450: 04 01 05 01 1F 55 25 25 A8 15 00 01 02 00 00 00 :89
C460: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C470: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C480: 23 3F 5F 5F 36 31 11 2C 00 1F 1F 1F 9F 1A 11 1A :05
C490: 0C 05 01 09 01 0B A7 0B F8 15 00 01 02 00 00 00 :E9
C4A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C4B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C4C0: 04 58 60 54 10 1F 1F 1B 00 92 92 92 92 02 02 02 :C7
C4D0: 02 00 00 00 00 07 05 07 06 15 15 00 01 02 00 00 :48
C4E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C4F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----
sum : AD DE 4F 43 E3 DB 27 99 CC 53 86 1C C9 26 29 28 :9C

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: 07 01 03 02 04 00 00 00 00 1F 1F 1F 1F 00 00 00 :8D
C510: 00 00 00 00 00 1F 1F 1F 1F 15 15 15 15 00 01 02 :D3
C520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C540: 3A 02 02 26 12 19 2A 20 00 8D 15 4F 52 06 07 08 :31
C550: 04 02 00 00 00 18 28 18 28 15 00 01 02 00 00 00 :9E
C560: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C570: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C580: 3A 01 01 2A 11 1A 36 1E 00 8E 4F 59 52 0A 0B 0D :8F
C590: 03 00 00 03 00 15 25 FF 28 15 00 01 02 00 00 00 :7F
C5A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C5B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C5C0: 3A 01 01 07 01 17 26 28 00 8D 8E 8D 53 0E 0E 0E :CE
C5D0: 03 00 00 00 00 13 13 FA 0A 15 00 01 02 00 00 00 :45
C5E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C5F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----
sum : BF 07 07 5C 28 A9 05 96 79 1B 26 6C 31 1E 21 25 :50

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: 3A 11 01 15 11 1D 30 0F 00 59 5C 59 4E 0A 0D 0B :4C
C610: 04 00 00 00 00 15 26 58 06 15 00 01 02 00 00 00 :B5
C620: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C640: 3A 12 02 1A 12 1B 33 18 00 5C 5C 53 4D 07 07 09 :4F
C650: 04 00 00 00 00 13 13 32 07 15 00 01 02 00 00 00 :7B

```



C660:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C670:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C680:	14	11	0A	21	11	17	39	0A	00	9A	DA	98	D8	0F	07	0C	:C1
C690:	0C	00	03	05	05	26	46	28	28	15	15	00	01	02	00	00	:02
C6A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C6B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C6C0:	1C	3F	01	01	01	28	2A	14	00	9F	9E	DB	5E	0F	06	07	:56
C6D0:	06	08	0B	0A	00	8A	F6	86	F7	15	15	00	01	02	00	00	:4D
C6E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C6F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	BE	7B	1C	60	3A	4F	3B	7D	2C	42	5A	21	D7	33	21	27	:31
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C700:	3A	31	64	0C	01	21	31	47	00	9C	9C	DB	DA	04	04	09	:73
C710:	03	03	03	01	00	17	06	02	A5	15	00	01	02	00	00	00	:E6
C720:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C730:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C740:	39	32	33	33	31	29	2B	23	00	DF	DF	DF	9F	07	04	04	:C4
C750:	0A	04	04	04	03	F7	FB	FB	0B	15	00	01	02	00	00	00	:29
C760:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C770:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C780:	20	36	30	35	31	1C	16	3A	00	DF	9F	DF	9F	07	09	06	:6A
C790:	06	07	06	06	08	29	19	19	F9	15	00	01	02	00	00	00	:8D
C7A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C7B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C7C0:	30	30	30	30	30	17	17	3F	00	9E	DC	D8	DC	0E	04	0A	:A7
C7D0:	05	08	08	08	08	B6	B6	B6	B6	15	00	01	02	00	00	00	:15
C7E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C7F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	DB	DF	0C	B7	A6	6A	59	AF	5F	4C	F6	75	FC	20	15	1D	:F9
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C800:	3D	35	52	78	31	27	1B	11	00	DF	1F	1F	1F	12	0F	04	:21
C810:	0F	00	00	00	00	2F	0F	0F	0F	15	15	15	00	01	02	00	:AD
C820:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C830:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C840:	1F	66	53	31	22	11	0A	0A	00	1C	1F	9F	1F	12	0F	0F	:79
C850:	0F	00	00	00	00	FF	0F	0F	0F	15	15	15	15	00	01	02	:92
C860:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C870:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C880:	04	58	60	54	10	1F	1F	1B	00	94	92	90	92	02	02	02	:C7
C890:	02	00	00	00	00	07	05	07	06	15	15	00	01	02	00	00	:48
C8A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C8B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C8C0:	1C	14	0C	12	06	26	26	19	00	92	5E	12	59	0A	06	04	:28
C8D0:	05	02	02	00	00	47	19	08	09	15	15	00	01	02	00	00	:A7
C8E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C8F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	A1	09	13	0F	69	F9	A6	7C	2D	75	82	8A	40	35	29	1B	:B7
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C900:	3B	56	33	06	03	28	26	33	00	DF	D0	54	8F	09	0B	07	:FB
C910:	04	03	00	00	00	E5	25	F5	08	15	00	01	02	00	00	00	:26
C920:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C930:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C940:	3E	04	04	04	04	23	0A	0C	00	1F	14	14	14	0A	0B	06	:FD
C950:	0B	0F	00	00	00	59	19	F9	19	15	15	15	00	01	02	00	:E0
C960:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C970:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C980:	3A	31	32	39	34	1F	32	28	00	D9	56	DC	54	0B	00	0C	:F9
C990:	06	00	00	0C	00	13	1B	5B	0B	15	00	01	02	00	00	00	:BE
C9A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C9B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00



C9C0:	3B	01	04	02	01	27	33	24	00	5F	5F	13	53	00	0F	0B	:FF
C9D0:	0F	00	00	00	00	00	09	4B	09	15	00	01	02	00	00	00	:84
C9E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C9F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	12	9E	6D	51	3C	E2	F7	1F	35	8A	AE	6F	50	1F	27	24	:38
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CA00:	1C	3B	07	01	51	21	17	15	00	DF	5F	DE	DE	0E	10	09	:1E
CA10:	07	00	05	07	04	FF	A0	16	17	15	15	00	01	02	00	00	:10
CA20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CA30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CA40:	24	39	05	76	01	1E	41	19	00	5F	9E	DB	9E	10	0C	07	:EA
CA50:	05	00	0B	0A	0A	BA	F6	85	F5	15	15	00	01	02	00	00	:7B
CA60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CA70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CA80:	3C	35	33	21	31	19	1B	0C	00	59	59	59	59	17	14	0E	:D3
CA90:	0E	0A	0B	0B	0C	E8	E8	F8	F8	15	15	00	01	02	00	00	:27
CAA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CAB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CAC0:	38	33	31	34	31	1E	1E	1E	00	DA	DC	DD	DF	08	04	05	:DE
CAD0:	0A	05	02	04	03	27	36	14	15	15	00	01	02	00	00	00	:B6
CAE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CAF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	D8	EB	8D	EC	D1	3E	45	FF	19	C5	71	F0	B9	43	34	23	:21
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CB00:	02	37	21	39	31	1C	1D	45	00	9F	9F	9F	9F	06	0C	06	:D6
CB10:	0C	06	06	06	06	01	01	01	05	15	00	01	02	00	00	00	:44
CB20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CB30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CB40:	3B	3C	30	39	31	21	07	22	00	DF	1F	1F	DF	04	04	05	:64
CB50:	01	04	04	04	02	F7	17	07	AC	15	00	01	02	00	00	00	:E8
CB60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CB70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CB80:	3A	0C	01	1F	73	20	1E	39	00	1F	1F	DF	9F	0C	0C	02	:26
CB90:	05	04	04	04	07	1A	06	F6	27	15	00	01	02	00	00	00	:6D
CBA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CBB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CBC0:	1C	33	35	21	33	16	19	0F	00	1D	CA	1D	5E	04	04	04	:84
CBD0:	04	01	04	03	03	20	30	03	03	15	15	00	01	02	00	00	:92
CBE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CBF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	A9	C1	99	C3	1A	A5	A9	B0	DB	0E	BC	BD	82	1C	20	11	:0F
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CC00:	39	02	02	01	01	1C	30	2A	00	5F	1F	5F	9F	0C	06	0D	:50
CC10:	0C	00	05	06	07	F4	15	15	14	15	00	01	02	00	00	00	:68
CC20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CC30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CC40:	3C	31	3B	31	31	1A	24	0A	00	10	1F	12	1F	06	0F	03	:CA
CC50:	0F	01	06	06	04	03	F2	0C	F2	15	15	00	01	02	00	00	:40
CC60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CC70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CC80:	38	3A	11	0A	02	28	1D	33	00	14	10	14	0E	05	02	08	:5C
CC90:	08	00	00	00	00	99	09	09	19	15	00	01	02	00	00	00	:E4
CCA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CCB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CCC0:	04	53	38	03	33	23	1C	0A	00	56	54	99	99	05	07	0C	:02
CCD0:	0C	06	06	06	06	64	64	15	65	15	15	00	01	02	00	00	:93
CCE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CCF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	E0	C7	97	51	78	75	01	B0	84	2D	CC	20	6B	20	1E	24	:97



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CD00:	39	55	55	57	05	16	25	2A	00	1D	1D	1D	1F	06	0D	00	:2D
CD10:	00	00	06	06	07	F4	24	04	05	15	00	01	02	00	00	00	:4C
CD20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CD30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CD40:	3C	35	34	39	30	0A	29	15	00	4D	0C	CD	14	02	01	02	:95
CD50:	01	02	01	02	01	08	56	08	18	15	15	00	01	02	00	00	:B2
CD60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CD70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CD80:	3E	02	02	02	01	30	1B	1D	00	04	04	04	00	00	00	00	:B9
CD90:	00	02	02	02	02	05	05	05	05	15	15	15	00	01	02	00	:5E
CDA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CDB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CDC0:	3E	38	32	32	32	31	53	25	00	DF	DF	D4	15	0A	0B	06	:77
CDD0:	0B	0F	0F	00	09	59	19	F9	19	15	15	15	00	01	02	00	:F8
CDE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CDF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	FD	D7	D5	CE	7B	DB	54	8B	3B	A1	4B	ED	4B	16	1D	08	:46

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CE00:	3C	3F	37	21	31	29	30	16	00	D2	DE	DE	DE	0C	0E	0F	:08
CE10:	0A	0A	0B	0B	01	F7	E7	F7	17	15	15	00	01	02	00	00	:44
CE20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CE30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CE40:	34	01	01	01	01	1F	1F	00	00	10	10	10	10	01	01	02	:BA
CE50:	02	00	00	00	00	18	18	18	18	15	15	00	01	02	00	00	:8F
CE60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CE70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CE80:	3C	0F	00	00	00	0E	13	11	00	1F	1F	1F	1F	00	18	0F	:20
CE90:	13	00	00	11	10	08	B8	2C	2C	15	15	00	01	02	00	00	:79
CEA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CEB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CEC0:	3B	7F	07	78	02	00	20	23	00	1F	1F	1F	1F	15	15	15	:39
CED0:	13	13	0D	0C	10	26	36	26	29	15	00	01	02	00	00	00	:12
CEE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CEF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	19	EB	57	C2	55	93	6F	AB	84	74	6B	2D	31	28	3C	35	:79

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
CF00:	34	00	50	0E	11	0D	1D	14	00	1F	1F	1F	5F	01	16	0D	:C1
CF10:	14	00	07	0F	14	25	68	FA	F8	15	15	00	01	02	00	00	:EA
CF20:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF30:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF40:	34	07	5E	05	3A	17	16	0A	00	1E	DF	1F	DF	13	13	0E	:3E
CF50:	0F	1E	0D	1C	0A	D8	4C	1B	19	15	15	00	01	02	00	00	:E5
CF60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF80:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF90:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFC0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFD0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CFF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	8B	25	C2	3E	69	21	E7	33	11	67	28	3E	40	18	29	1B	:CE



行番号	説	明
10	音色データのトップアドレス入力	AD\$
20	AD\$を、16進数の数にします	AD
30	アドレストップを、保存します	AF = AD + 1
40	FM音源内のレジスタ番号を作ります	RE
50	メモリからデータをもってきます。メモリを一つ足します	DA
60	OPNにレジスタ番号を出力	RE
70	OPNのレジスタに、データを出力	DA
80	レジスタ番号とデータを表示	
90	ループ	
100	} OPNのチャンネル1の四つのオペレーターのつなぎ方を指定します	
110		
120	OPNのおまじない (イニシャライズ)	
130	} 音を出すときの音程を指定します (このプログラムではオクターブ4のド)	
140		
150		
160		
190	OPNチャンネル1 Key on!	
200	タイマー	
210	key off	
220	おしまい	

- OPNのどのレジスタに、メモリのどのデータが出力されたかは、プログラムを実行すれば分かります。
- 実は、このプログラム中でも音程設定しています。

### 第3-2-2図 音色設定プログラム(BASIC部)説明

- ひとまとまりのデータの一番先頭の1バイトは、OPNのレジスタ0B0Hのコネクションというレジスタに出力するデータです。
- その次のデータから、OPN内レジスタ30Hから4ステップで出力するデータです。

《例. C0C0Hからの場合》

C0C0H	3AH	OPNレジスタ	B0Hのデータ
C0C1H	3IH	OPNレジスタ	30Hのデータ
C0C2H	64H	OPNレジスタ	34Hのデータ
C0C3H	0CH	OPNレジスタ	38Hのデータ
C0C4H	0IH	OPNレジスタ	3CHのデータ
C0C5H	2IH	OPNレジスタ	40Hのデータ
C0C6H	3IH	OPNレジスタ	44Hのデータ
⋮	⋮	⋮	⋮
C0DDH	00H	OPNレジスタ	A0Hのデータ

### 第3-2-3図 音色データの見方



まで、すべて順番どおりに入っています。

音色番号とメモリアドレスの関係を式にすると次のようになります。

**アドレス=&HC000+&H40×音色番号**

最後に、このBASICの音色設定プログラムをマシン語にしたものが、リスト3-2-3です(リスト3-2-4がソースリスト)。

リスト3-2-3の使い方は、リストを正しく打ち込み、SAVEした後、まず先ほどのリスト3-2-2のマシン語リストを見て下さい。

そして、自分が設定をしたい音色のデータをOPNチャンネルによって、

チャンネル1ならD0A0Hから、

チャンネル2ならD0C0Hから、

チャンネル3ならD0E0Hから、

それぞれ32バイト入力して下さい。

チャンネル1や2,3の意味が分からない人、この意味は、

CMD PLAY A\$,B\$,C\$

で、A\$を演奏するのがチャンネル1で、B\$を演奏するのがチャンネル2、そしてC\$を演奏するのがチャンネル3です。

音色番号と、音色データ・メモリアドレスの関係は、さきほどの式を使用して下さい。

さあ、音色データを3チャンネルともすべて入力しましたか？ そうしたら、BASICで

DEF USR=&HD000

I=USR(0)

を実行しましょう。さあこれでOPNに音色データがセットされました。

では、OPNのKeyをonしてみてください。

チャンネル1のonは、

OUT &H44,&H28

OUT &H45,&HF0

です。チャンネル2・3のonは、それぞれ上の

&HF0を、

&HF1——→チャンネル2のon

&HF2——→チャンネル3のon

とすれば、よいわけです。

Key onして、スピーカーから音が出ましたか？

たぶん“出ない”又は、“ノイズのような音”でしょう。なぜでしょうねえ。プログラムがバグっているのでしょうか？

答えは、

“FM音源をイニシャライズしないまま

音色を設定した！”

“FM音源に音程をセットしていない！”

からです。



「オレの88は音でたよー。」

と言っている人。あなたはこのプログラムを実行する前に、PLAY文を使ったでしょう？ 又は、BASICの音色設定プログラムを実行した後、一度もリセットや電源を切っていないでしょう。

その人たちの88は、OPNをイニシャライズして、音程設定をしてあるから音が出たのです。

さあ、それでは次に、FM音源のイニシャライズと、音程設定についてお話ししましょう。

### リスト3-2-3 音色設定プログラム(BASIC部をマシン語に変換)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D000:	CD	10	D0	CD	40	D0	CD	70	D0	C9	06	00	10	FE	C9	00	:3D
D010:	21	A0	D0	3E	B0	D3	44	CD	0A	D0	7E	D3	45	CD	0A	D0	:7A
D020:	23	11	30	00	06	1C	C5	7B	D3	44	CD	0A	D0	7E	D3	45	:1A
D030:	1C	1C	1C	1C	23	CD	0A	D0	C1	10	EB	C9	00	00	00	00	:BF
D040:	21	C0	D0	3E	B1	D3	44	CD	0A	D0	7E	D3	45	CD	0A	D0	:9B
D050:	23	11	31	00	06	1C	C5	7B	D3	44	CD	0A	D0	7E	D3	45	:1B
D060:	1C	1C	1C	1C	23	CD	0A	D0	C1	10	EB	C9	00	00	00	00	:BF
D070:	21	E0	D0	3E	B2	D3	44	CD	0A	D0	7E	D3	45	CD	0A	D0	:BC
D080:	23	11	32	00	06	1C	C5	7B	D3	44	CD	0A	D0	7E	D3	45	:1C
D090:	1C	1C	1C	1C	23	CD	0A	D0	C1	10	EB	C9	00	00	00	00	:BF
D0A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D0B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D0C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D0D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D0E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D0F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	ED	D7	27	DB	CE	04	06	B8	AA	35	A8	F2	4F	DF	60	3F	:9C

### リスト3-2-4 音色設定プログラム・ソースリスト

```

;
; マシンコ" オンショク セッテイ ルーチン
;
;
;          ORG  0D000H
;
;
D000 CD10D0          CALL CH1
D003 CD40D0          CALL CH2
D006 CD70D0          CALL CH3
D009 C9              RET
;
D00A 0600          TIMER: LD  B,0
D00C 10FE          LOOP1: DJNZ LOOP1
D00E C9              RET
;
D00F 00              NOP
D010 21A0D0          CH1:  LD  HL,CH1DATA          ;CH1 SET DATA TOP ADDRESS
D013 3EB0              LD  A,0B0H

```



```

D015 D344          OUT  (44H),A          ;OPN REG SET
D017 CD0AD0        CALL TIMER
D01A 7E            LD   A,(HL)
D01B D345          OUT  (45H),A          ;OPN DATA SET
D01D CD0AD0        CALL TIMER
D020 23            INC  HL
D021 113000        LD   DE,30H

;
D024 061C          LD   B,1CH
D026 C5            CH1LP: PUSH BC
D027 7B            LD   A,E
D028 D344          OUT  (44H),A          ;OPN REG SET
D02A CD0AD0        CALL TIMER
D02D 7E            LD   A,(HL)
D02E D345          OUT  (45H),A          ;OPN DATA SET
D030 1C            INC  E
D031 1C            INC  E
D032 1C            INC  E
D033 1C            INC  E
D034 23            INC  HL
D035 CD0AD0        CALL TIMER
D038 C1            POP  BC
D039 10EB          DJNZ CH1LP

;
D03B C9            RET

;
D03C 00            NOP
D03D 00            NOP
D03E 00            NOP
D03F 00            NOP
D040 21C0D0        CH2: LD   HL,CH2DATA      ;CH2 SET DATA TOP ADDRESS
D043 3EB1          LD   A,0B1H
D045 D344          OUT  (44H),A          ;OPN REG SET
D047 CD0AD0        CALL TIMER
D04A 7E            LD   A,(HL)
D04B D345          OUT  (45H),A          ;OPN DATA SET
D04D CD0AD0        CALL TIMER
D050 23            INC  HL
D051 113100        LD   DE,31H

;
D054 061C          LD   B,1CH
D056 C5            CH2LP: PUSH BC
D057 7B            LD   A,E
D058 D344          OUT  (44H),A          ;OPN REG SET
D05A CD0AD0        CALL TIMER
D05D 7E            LD   A,(HL)
D05E D345          OUT  (45H),A          ;OPN DATA SET
D060 1C            INC  E
D061 1C            INC  E
D062 1C            INC  E
D063 1C            INC  E
D064 23            INC  HL
D065 CD0AD0        CALL TIMER
D068 C1            POP  BC
D069 10EB          DJNZ CH2LP

;
D06B C9            RET

;
D06C 00            NOP
D06D 00            NOP
D06E 00            NOP
D06F 00            NOP
D070 21E0D0        CH3: LD   HL,CH3DATA      ;CH3 SET DATA TOP ADDRESS
D073 3EB2          LD   A,0B2H
D075 D344          OUT  (44H),A          ;OPN REG SET
D077 CD0AD0        CALL TIMER
D07A 7E            LD   A,(HL)
D07B D345          OUT  (45H),A          ;OPN DATA SET
D07D CD0AD0        CALL TIMER
D080 23            INC  HL

```



```

D081 113200      LD    DE,32H
;
D084 061C        LD    B,1CH
D086 C5          CH3LP: PUSH BC
D087 7B          LD    A,E
D088 D344        OUT   (44H),A      ;OPN REG SET
D08A CD0AD0      CALL  TIMER
D08D 7E          LD    A,(HL)
D08E D345        OUT   (45H),A      ;OPN DATA SET
D090 1C          INC    E
D091 1C          INC    E
D092 1C          INC    E
D093 1C          INC    E
D094 23          INC    HL
D095 CD0AD0      CALL  TIMER
D098 C1          POP    BC
D099 10EB        DJNZ  CH3LP
;
D09B C9          RET
;
D09C 00          NOP
D09D 00          NOP
D09E 00          NOP
D09F 00          NOP
D0A0             CH1DATA
D0A0             DS    20H          ;チャンネル 1 データ 32 バイト
;
;
D0C0             CH2DATA
D0C0             DS    20H          ;チャンネル 2 データ 32 バイト
;
;
D0E0             CH3DATA
D0E0             DS    20H          ;チャンネル 3 データ 32 バイト
;
;
D100             END

```

### 3-2-5

## OPNのイニシャライズ

PC-88では、電源を入れてBASICを起動しても、FM音源LSI, OPNのイニシャライズはしません。

### NEW CMD

を実行して初めてOPNがイニシャライズされます。

### CMD STOPM

は、再びイニシャライズするときに使うコマンドです。

OUT命令でOPNをイニシャライズするのは、とっても簡単で、OPNのレジスタ&H2Dを指定するだけです。つまり、

OUT &H44,&H2D

です。レジスタを指定するだけで、データは出力しません。

### 3-2-6

## FM音源の音程設定



さあ、FM音源のイニシャライズの方法、

FM音源の音色設定するプログラム、

FM音源のkey on / offの方法

が分かりました。あとは音程設定すれば、PLAY文を使わなくても自由に音を出すことができます。

音程の設定は、OPNレジスタの

&HA0～&HA6

を使用します。このレジスタは次のようになっています。

チャンネル1     &HA4と&HA0

チャンネル2     &HA5と&HA1

チャンネル3     &HA6と&HA2

です。詳しい説明は、**まとめ** を御覧下さい。レジスタの内容は、図を見てもらえば分かるので、ここでは具体的な数値で話をしていきます。

まず、一番よく聞く音、オクターブ4のドの音は、

CMD PLAY "04C"

となりますが、この音程をレジスタへ出力する数値にすると、

&H1Aと&H69になります。

第3-2-1表 音程と数値の関係

●OPNレジスタ     &HA4……チャンネル1  
                          &HA5……チャンネル2  
                          &HA6……チャンネル3

に、出力する値

音 オクターブ	ド	ド#	レ	レ#	ミ	ファ	ファ#	ソ	ソ#	ラ	ラ#	シ
1	02H	02H	02H	02H	03H	03H	03H	03H	03H	04H	04H	04H
2	0AH	0AH	0AH	0AH	0BH	0BH	0BH	0BH	0BH	0CH	0CH	0CH
3	12H	12H	12H	12H	13H	13H	13H	13H	13H	14H	14H	14H
4	1AH	1AH	1AH	1AH	1BH	1BH	1BH	1BH	1BH	1CH	1CH	1CH
5	22H	22H	22H	22H	23H	23H	23H	23H	23H	24H	24H	24H
6	2AH	2AH	2AH	2AH	2BH	2BH	2BH	2BH	2BH	2CH	2CH	2CH
7	32H	32H	32H	32H	33H	33H	33H	33H	33H	34H	34H	34H
8	3AH	3AH	3AH	3AH	3BH	3BH	3BH	3BH	3BH	3CH	3CH	3CH

●OPNレジスタ     &HA0……チャンネル1  
                          &HA1……チャンネル2  
                          &HA2……チャンネル3

に、出力する値は、どのオクターブも共通です。

ド	ド#	レ	レ#	ミ	ファ	ファ#	ソ	ソ#	ラ	ラ#	シ
69H	8FH	B5H	DFH	0AH	38H	6AH	9DH	D5H	0EH	4CH	8DH



「なんでこういう数なのー？」

と、疑問に思うかもしれませんが、実はこの値は、

ド——→1

ドの#→ 2

レ——→3

などという、半音きざみの数値ではなくて、

“発音させる音の周波数に関係している値”

なのです。

従って、ここではこの数値については、くどくど書きません。しかし、ドレミファソラシとそれぞれの半音と、レジスタへ出力する数値の関係をまとめた表を第3-2-1表に示しました。

この表の見方は、例えば、オクターブ6のファのシャープ(#)の音をチャンネル2を使って出したいのならば、

OUT &H44,&HA5 □ レジスタ指定

OUT &H45,&H2B □ データ出力

それと、

OUT &H44,&HA1 □

OUT &H45,&H6A □

とすればよいのです。このとき、OPNレジスタの

&HA4~&HA6

に出力することを必ず先にして下さい。

&HA0~&HA2

の出力を先にしてしまうと、音が変な音になってしまいます。

第3-2-1表の数値をいろいろいじくれば、半音の半音とか、“ド”の音ではないけれど、それに聞こえる音などが作れるでしょう。

## 3-2-7

### まとめ

#### ●FM音源LSI

I/O &H44  
&H45

PC-88のI/Oポートの&H44にOPNのレジスタを、&H45にデータを出力します。

#### ●Key on / off

OPNレジスタ &H28

このレジスタで、各チャンネルのKey on / offを指示します。

#### ●音色設定

リスト3-2-2  
リスト3-2-3



●イニシャライズ

OPNレジスタ &H2D  
OUT &H44,&H2D  
だけです。

●音程設定

OPNレジスタ &HA4～&HA6, &HA0～&HA2  
このレジスタに、第3-2-1表の数値を出力して音程を指定します。

まとめ1●本章で登場したOPN内のレジスタ

OPN レジスタ番号

28H key on/off コントロール

出力データ

ビット 7	ビット 6	ビット 5	ビット 4		ビット 1	ビット 0
----------	----------	----------	----------	--	----------	----------

ビット 1	ビット 0	チャンネル
0	0	チャンネル 1
0	1	チャンネル 2
1	0	チャンネル 3

チャンネルとは  
CMD, PLAY チャンネル 1, チャンネ  
ル 2, チャンネル 3 のことです。

○この 2 ビットで、key on/off するチャンネルを指定する。

ビット 4	オペレーター 1 の on/off
ビット 5	オペレーター 2 の on/off
ビット 6	オペレーター 3 の on/off
ビット 7	オペレーター 4 の on/off

○この 4 ビットで、上で指定したチャンネルの四つのオペレーターを、on/off します。

ビットが 1 のとき on  
0 のとき off です。  
普通はすべて on 又は off にします。

まとめ2●FM音源の音程設定

音程を設定するには、オクターブ(1~8)と、そのオクターブの中での音程を設定します。

- オクターブ 3 ビット
- オクターブ中の音程 11ビット



## OPNレジスタ

## 下位の8ビット

1 ……A0H							
チャンネル 2 ……A1H	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1
3 ……A2H							ビット 0

## OPNレジスタ

## 上位の3ビット

1 ……A4H			
チャンネル 2 ……A5H	ビット 2	ビット 1	ビット 0
3 ……A6H			

○この11ビットで、オクターブ中の周波数に関係した値を設定する。

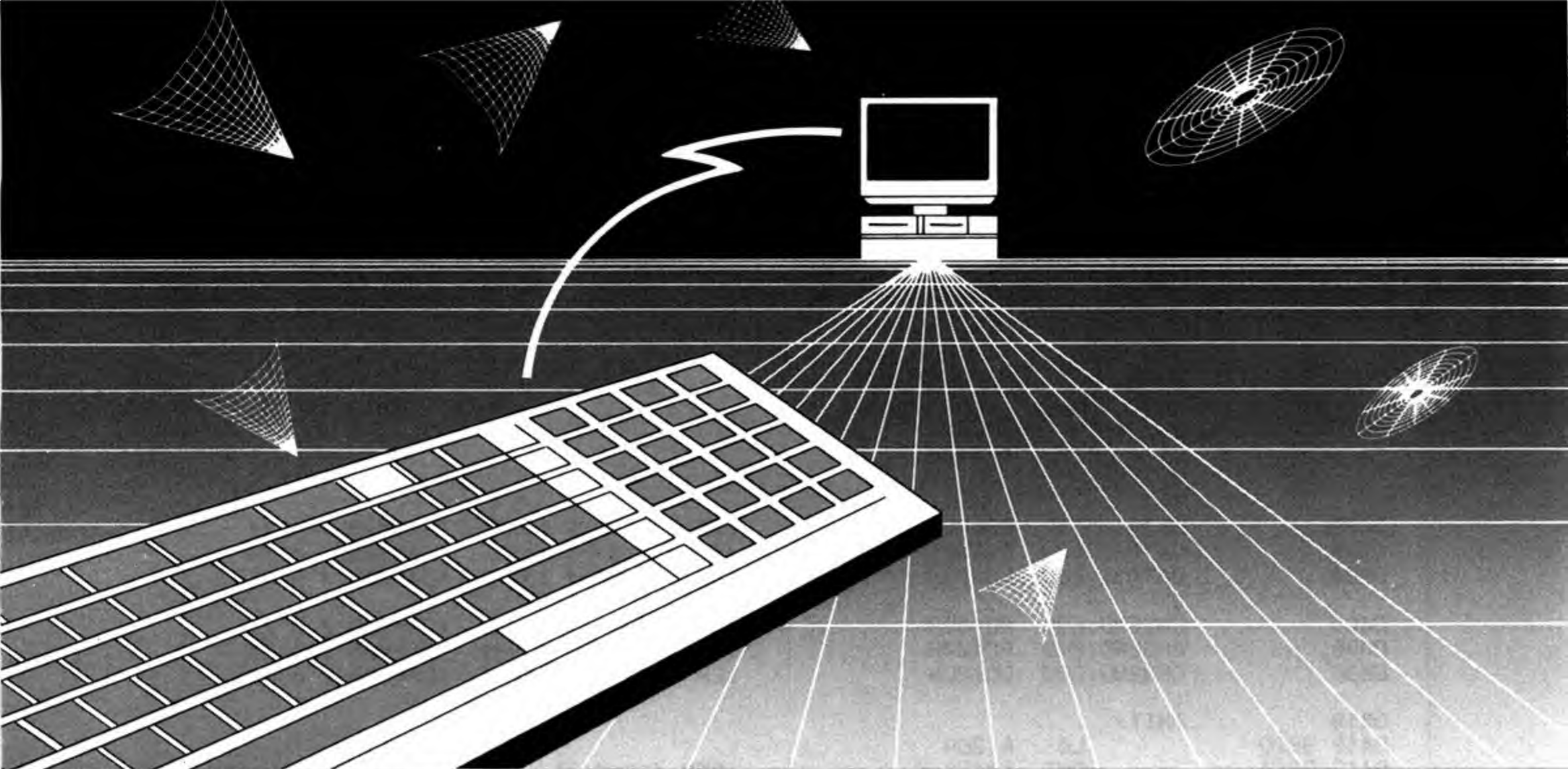
## OPN内レジスタ

1 ……A4H			
チャンネル 2 ……A5H	ビット 5	ビット 4	ビット 3
3 ……A6H			

○この3ビットで、オクターブを設定する。

	ビット 5	・	4	・	3
CMD PLAY "O1"	→ 0		0		0
CMD PLAY "O2"	0		0		1
CMD PLAY "O3"	0		1		0
CMD PLAY "O4"	0		1		1
CMD PLAY "O5"	1		0		0
CMD PLAY "O6"	1		0		1
CMD PLAY "O7"	1		1		0
CMD PLAY "O8"	1		1		1





## 3-3 FM音源BGMルーチン

### 3-3-1

#### 集大成！FM音源BGMルーチン

やっこここまでたどりつきました。今まで説明してきたことを一つにして、割り込み利用のFM音源BGMルーチンを御披露いたしましょう。

リスト3-3-1(アセンブルリスト)と、

リスト3-3-2(ダンプリスト)です。

このプログラムの入力の方法は、

CLEAR ,&HD3FF

と入力してから、ダンプリストをモニタで入力して下さい。

このダンプリストは、BGMルーチンのメインで、音程や音長のデータは入っていません。

そこで、コツコツと音楽データを入力していくわけですが、面倒ですねえ。いちいち音階と第3-2-1表を見ながらメモリに書き込んでいったら、すぐに嫌になってしまいます。

ここで、この割り込みルーチン専用のユーティリティの登場です。リスト3-3-3がそれです。BASICなので、解析するのは簡単でしょう。

次に、このユーティリティの使い方を説明します。

まず、リスト3-3-3を入力し、バグ取りしてディスクにSAVEします。次にCLEAR文でマシン語プログラムエリアを確保した後、さきほどBGMのメインプログラム(リスト3-3-2)を自分の好きなアドレスにロードして下さい。

例えば、

CLEAR ,&HBFFF

BLOAD "リスト3-3-2", &HC000などです。(リスト3-3-2は、本当は&HD400から入力しなくてもよかったのです)



## リスト3-3-1 FM音源BGMメインルーチン(アセンブルリスト)

```

:
:
:           BGM MAIN ルーチン
:           BY K.ISHIDA
:
:           ORG 0D410H
:
D600      DATA1: EQU 0D600H           ;CH 1 ノ テータ トッフ° (GAMMA5 ノ ハアイ
D700      DATA2: EQU 0D700H           ;CH 2 ノ テータ トッフ° (GAMMA5 ノ ハアイ
DA00      DATA3: EQU 0DA00H           ;CH 3 ノ テータ トッフ° (GAMMA5 ノ ハアイ
:
D6CC      CH1END: EQU 0D6CCH           ;CH 1 ノ テータ オクリ+1 (GAMMA5 ノ ハアイ
DA00      CH2END: EQU 0DA00H           ;CH 2 ノ テータ オクリ+1 (GAMMA5 ノ ハアイ
DB3C      CH3END: EQU 0DB3CH           ;CH 3 ノ テータ オクリ+1 (GAMMA5 ノ ハアイ
:
D410      INIT:
D410      LD A,2DH
D412      OUT (44H),A                 ;OPN イニシャライズ
D414      NOP
D415      NOP
D416      NOP
D417      NOP
D418      NOP
D419      NOP
D41A      NOP
D41B      NOP
D41C      NOP
D41D      NOP
D41E      NOP
D41F      NOP
D420      NOP
D421      LD A,0D4H
D423      LD I,A                      ;I レジスタ セッテイ
D425      LD A,3
D427      OUT (0E4H),A                ;ワリコミ コントローラー レベル セッテイ
D429      LD HL,MAIN
D42C      LD (0D404H),HL              ;ワリコミ テーブル ニ カキコム
D42F      LD HL,0000H
D432      LD (NCH1),HL
D435      LD (NCH2),HL                ;チャンネル コートノ オトノ ナカサ リセット
D438      LD (NCH3),HL
D43B      LD HL,DATA1
D43E      LD (CH1CN),HL
D441      LD HL,DATA2
D444      LD (CH2CN),HL
D447      LD HL,DATA3
D44A      LD (CH3CN),HL
D44D      EI
D44E      RET
;
D44F      NOP
D450      MAIN: DI                     ;CPU ワリコミ タメ
D451      PUSH BC
D452      PUSH DE
D453      PUSH HL                     ;CPU ノ レジスタ ホソソ
D454      PUSH AF
D455      CALL FMPLAY
D458      CALL PSGPLAY
D45B      LD A,3
D45D      OUT (0E4H),A                ;ワリコミ コントローラー レベル セッテイ
D45F      POP AF
D460      POP HL
D461      POP DE                     ;CPU ノ レジスタ モトシ
D462      POP BC
D463      EI
D464      RET                         ;CPU ワリコミ Ok!
;                                     ;RETURN
D465      DATA4:
D465      DB 0EEH
D466      DB 077H
D467      DB 0BDH

```



```

D468 5E          DB 05EH
D469 9F          DB 09FH
D46A 4F          DB 04FH
D46B 7E          DB 07EH
D46C 3F          DB 03FH
D46D 77          DB 077H
D46E 3B          DB 03BH
D46F 00          DATA4E:DB 000H          ;シュツゲン オン イント
;
;
D470          FMPLAY:
D470 2AF0D5      LD HL,(NCH1)
D473 7C          LD A,H
D474 B5          OR L
D475 CCA0D4      CALL Z,CH1PLAY          ;IF CH1 ノ ナカ"サ=0 THEN ツキ"ノ オトダ"シ
D478 2AF0D5      LD HL,(NCH1)
D47B 2B          DEC HL
D47C 22F0D5      LD (NCH1),HL
D47F 00          NOP
;
D480 2AF4D5      LD HL,(NCH2)
D483 7C          LD A,H
D484 B5          OR L
D485 CCF0D4      CALL Z,CH2PLAY          ;IF CH2 ノ ナカ"サ=0 THEN ツキ"ノ オトダ"シ
D488 2AF4D5      LD HL,(NCH2)
D48B 2B          DEC HL
D48C 22F4D5      LD (NCH2),HL
D48F 00          NOP
;
D490 2AF8D5      LD HL,(NCH3)
D493 7C          LD A,H
D494 B5          OR L
D495 CC40D5      CALL Z,CH3PLAY          ;IF CH3 ノ ナカ"サ=0 THEN ツキ"ノ オトダ"シ
D498 2AF8D5      LD HL,(NCH3)
D49B 2B          DEC HL
D49C 22F8D5      LD (NCH3),HL
D49F C9          RET
;
D4A0          CH1PLAY:
D4A0 3E28        LD A,28H
D4A2 D344        OUT (44H),A
D4A4 3E00        LD A,0
D4A6 D345        OUT (45H),A          ;チャンネル 1 Key OFF
D4A8 2AF2D5      LD HL,(CH1CN)
D4AB 7E          LD A,(HL)
D4AC EE00        XOR 0
D4AE C2B9D4      JP NZ,ARI1
D4B1 23          INC HL
D4B2 7E          LD A,(HL)
D4B3 EE00        XOR 0
D4B5 CAD0D4      JP Z,NASHI1          ;オト ノ テ"ー タ カ" 2ツ トモ 00 ノ トキ ハ, オトナシ
D4B8 2B          DEC HL
D4B9          ARI1:
D4B9 3EA4        LD A,0A4H
D4BB D344        OUT (44H),A
D4BD 7E          LD A,(HL)
D4BE D345        OUT (45H),A          ;チャンネル 1 0A4H オンテイ セツテイ
D4C0 23          INC HL
D4C1 3EA0        LD A,0A0H
D4C3 D344        OUT (44H),A
D4C5 7E          LD A,(HL)
D4C6 D345        OUT (45H),A          ;チャンネル 1 0A0H オンテイ セツテイ
D4C8 3E28        LD A,28H
D4CA D344        OUT (44H),A
D4CC 3EF0        LD A,0F0H
D4CE D345        OUT (45H),A          ;チャンネル 1 Key ON!
D4D0          NASHI1:
D4D0 23          INC HL
D4D1 7E          LD A,(HL)
D4D2 32F0D5      LD (NCH1),A
D4D5 23          INC HL
D4D6 7E          LD A,(HL)
D4D7 32F1D5      LD (NCH1H),A          ;チャンネル 1 ノ オトノ ナカ"サ ラ カウンター ニ イレル
D4DA 23          INC HL
D4DB E5          PUSH HL
D4DC 01CCD6      LD BC,CH1END

```



```

D4DF A7          AND A
D4E0 ED42        SBC HL,BC
D4E2 E1          POP HL
D4E3 CCEAD4      CALL Z,CH1RESET
D4E6 22F2D5      LD (CH1CN),HL
D4E9 C9          RET

;
D4EA             CH1RESE
D4EA 2100D6      LD HL,DATA1
D4ED C9          RET

;
D4EE 00          NOP
D4EF 00          NOP
D4F0             CH2PLAY:
D4F0 3E28        LD A,28H
D4F2 D344        OUT (44H),A
D4F4 3E01        LD A,1
D4F6 D345        OUT (45H),A
D4F8 2AF6D5      LD HL,(CH2CN)
D4FB 7E          LD A,(HL)
D4FC EE00        XOR 0
D4FE C209D5      JP NZ,ARI2
D501 23          INC HL
D502 7E          LD A,(HL)
D503 EE00        XOR 0
D505 CA20D5      JP Z,NASHI2
D508 2B          DEC HL
D509             ARI2:
D509 3EA5        LD A,0A5H
D50B D344        OUT (44H),A
D50D 7E          LD A,(HL)
D50E D345        OUT (45H),A
D510 23          INC HL
D511 3EA1        LD A,0A1H
D513 D344        OUT (44H),A
D515 7E          LD A,(HL)
D516 D345        OUT (45H),A
D518 3E28        LD A,28H
D51A D344        OUT (44H),A
D51C 3EF1        LD A,0F1H
D51E D345        OUT (45H),A
D520             NASHI2:
D520 23          INC HL
D521 7E          LD A,(HL)
D522 32F4D5      LD (NCH2),A
D525 23          INC HL
D526 7E          LD A,(HL)
D527 32F5D5      LD (NCH2H),A
D52A 23          INC HL
D52B E5          PUSH HL
D52C 0100DA      LD BC,CH2END
D52F A7          AND A
D530 ED42        SBC HL,BC
D532 E1          POP HL
D533 CC3AD5      CALL Z,CH2RESET
D536 22F6D5      LD (CH2CN),HL
D539 C9          RET

;
D53A             CH2RESE
D53A 2100D7      LD HL,DATA2
D53D C9          RET

;
D53E 00          NOP
D53F 00          NOP
D540             CH3PLAY:
D540 3E28        LD A,28H
D542 D344        OUT (44H),A
D544 3E02        LD A,2
D546 D345        OUT (45H),A
D548 2AFAD5      LD HL,(CH3CN)
D54B 7E          LD A,(HL)
D54C EE00        XOR 0
D54E C259D5      JP NZ,ARI3
D551 23          INC HL
D552 7E          LD A,(HL)
D553 EE00        XOR 0

```

; IF チャンネル 1 ノ テーマ = END THEN テーマ  
 ;チャンネル 2 ノ オトダシ サブ  
 ;チャンネル 2 Key OFF  
 ;オト ノ テーマ カ 2ツ トモ 00 ノ トキ ハ, オトナシ  
 ;チャンネル 2 0A5H オンティ セッティ  
 ;チャンネル 2 0A1H オンティ セッティ  
 ;チャンネル 2 Key ON!  
 ;チャンネル 2 ノ オトノ ナカサヲ カウンター ニ イレル  
 ; IF チャンネル 2 ノ テーマ = END THEN テーマ  
 ;チャンネル 3 ノ オトダシ サブ  
 ;チャンネル 3 Key OFF



```

D555 CA70D5      JP      Z,NASHI3      ;オトノデータカ 2ツトモ 00ノトキハ、オトナシ
D558 2B          DEC      HL
D559             ARI3:
D559 3EA6        LD       A,0A6H
D55B D344        OUT      (44H),A
D55D 7E          LD       A,(HL)
D55E D345        OUT      (45H),A      ;チャンネル 3 0A6H オンテイ セッテイ
D560 23          INC      HL
D561 3EA2        LD       A,0A2H
D563 D344        OUT      (44H),A
D565 7E          LD       A,(HL)
D566 D345        OUT      (45H),A      ;チャンネル 3 0A2H オンテイ セッテイ
D568 3E28        LD       A,28H
D56A D344        OUT      (44H),A
D56C 3EF2        LD       A,0F2H      ;チャンネル 3 Key ON!
D56E D345        OUT      (45H),A
D570             NASHI3:
D570 23          INC      HL
D571 7E          LD       A,(HL)
D572 32F8D5      LD       (NCH3),A
D575 23          INC      HL
D576 7E          LD       A,(HL)
D577 32F9D5      LD       (NCH3H),A      ;チャンネル 3 ノ オトノ ナカサヲ カウンター ニ イレル
D57A 23          INC      HL
D57B E5          PUSH     HL
D57C 013CDB      LD       BC,CH3END
D57F A7          AND      A
D580 ED42        SBC      HL,BC
D582 E1          POP      HL
D583 CC8AD5      CALL     Z,CH3RESET      ;IF チャンネル 3 ノ データ = END THEN データ
D586 22FAD5      LD       (CH3CN),HL
D589 C9          RET
;
D58A             CH3RESE
D58A 2100DA      LD       HL,DATA3
D58D C9          RET
;
D58E 00          NOP
D58F 00          NOP
D590             PSGPLAY
D590 CDA0D5      CALL     PSGON
D593 3A9FD5      LD       A,(NPSG)
D596 3C          INC      A
D597 329FD5      LD       (NPSG),A
D59A C9          RET
;
D59B 00          NOP
D59C 00          NOP
D59D             PSGCN:
D59D 65D4        DW       DATA4
D59F             NPSG:
D59F 00          DB       00
;
D5A0             PSGON:
D5A0 3A9FD5      LD       A,(NPSG)
D5A3 EE08        XOR      8
D5A5 C0          RET      NZ      ;IF PSG ノ ナカサ カ<8 THEN RETURN
D5A6 3E00        LD       A,0
D5A8 329FD5      LD       (NPSG),A      ;PSG ノ ナカサ RESET
D5AB 3E07        LD       A,7
D5AD D344        OUT      (44H),A      ;レジスタ 7
D5AF 3E38        LD       A,38H
D5B1 D345        OUT      (45H),A      ;PSG モード セッテイ
D5B3 2A9DD5      LD       HL,(PSGCN)
D5B6 3E08        LD       A,8
D5B8 D344        OUT      (44H),A
D5BA 3E0F        LD       A,0FH      ;PSG ホリューム (サイタイ)
D5BC D345        OUT      (45H),A
D5BE 3E01        LD       A,1
D5C0 D344        OUT      (44H),A
D5C2 3E00        LD       A,0
D5C4 D345        OUT      (45H),A      ;PSG ニ オンテイ ラ オクル
D5C6 D344        OUT      (44H),A
D5C8 7E          LD       A,(HL)
D5C9 D345        OUT      (45H),A      ;PSG ニ オンテイ ラ オクル

```



```

D5C8 23          INC HL
D5CC E5          PUSH HL
D5CD 016FD4      LD BC,DATA4E
D5D0 A7          AND A
D5D1 ED42        SBC HL,BC
D5D3 E1          POP HL
D5D4 CCDBD5      CALL Z,PSGRESET          ; IF PSG / テータ = END THEN RESET
D5D7 229DD5      LD (PSGCN),HL
D5DA C9          RET

;
D5DB             ; PSGRESE
D5DB 2165D4      LD HL,DATA4
D5DE 3EC9        LD A,0C9H
D5E0 329DD5      LD (PSGPLAY),A          ; 1 カイ ナラシタラ オシマイ
D5E3 3E07        LD A,7
D5E5 D344        OUT (44H),A
D5E7 3EFF        LD A,0FFH              ; PSG モート" セッテイ
D5E9 D345        OUT (45H),A
D5EB C9          RET

;
D5EC 00          NOP
D5ED 00          NOP
D5EE 00          NOP
D5EF 00          NOP
D5F0 00          NCH1: DB 00
D5F1 00          NCH1H: DB 00
D5F2 00D6        CH1CN: DW DATA1
;
D5F4 00          NCH2: DB 00
D5F5 00          NCH2H: DB 00
D5F6 00D7        CH2CN: DW DATA2
;
D5F8 00          NCH3: DB 00
D5F9 00          NCH3H: DB 00
D5FA 00DA        CH3CN: DW DATA3
;
D5FC 00          NOP
D5FD 00          NOP
D5FE 00          NOP
D5FF 00          NOP
D600             END

```

## リスト3-3-2 FM音源BGMメインルーチン(ダンプリスト)

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D400:	EA	E7	08	E8	50	D4	14	E8	14	E8	14	E8	1A	E8	20	E8	:E3
D410:	3E	2D	D3	44	00	00	00	00	00	00	00	00	00	00	00	00	:82
D420:	00	3E	D4	ED	47	3E	03	D3	E4	21	50	D4	22	04	D4	21	:9E
D430:	00	00	22	F0	D5	22	F4	D5	22	F8	D5	21	00	D6	22	F2	:CC
D440:	D5	21	00	D7	22	F6	D5	21	00	DA	22	FA	D5	FB	C9	00	:6A
D450:	F3	C5	D5	E5	F5	CD	70	D4	CD	90	D5	3E	03	D3	E4	F1	:93
D460:	E1	D1	C1	FB	C9	EE	77	BD	5E	9F	4F	7E	3F	77	3B	00	:14
D470:	2A	F0	D5	7C	B5	CC	A0	D4	2A	F0	D5	2B	22	F0	D5	00	:61
D480:	2A	F4	D5	7C	B5	CC	F0	D4	2A	F4	D5	2B	22	F4	D5	00	:BD
D490:	2A	F8	D5	7C	B5	CC	40	D5	2A	F8	D5	2B	22	F8	D5	C9	:E3
D4A0:	3E	28	D3	44	3E	00	D3	45	2A	F2	D5	7E	EE	00	C2	B9	:AB
D4B0:	D4	23	7E	EE	00	CA	D0	D4	2B	3E	A4	D3	44	7E	D3	45	:8B
D4C0:	23	3E	A0	D3	44	7E	D3	45	3E	28	D3	44	3E	F0	D3	45	:71
D4D0:	23	7E	32	F0	D5	23	7E	32	F1	D5	23	E5	01	CC	D6	A7	:83
D4E0:	ED	42	E1	CC	EA	D4	22	F2	D5	C9	21	00	D6	C9	00	00	:0C
D4F0:	3E	28	D3	44	3E	01	D3	45	2A	F6	D5	7E	EE	00	C2	09	:00
sum :	D2	56	BD	39	EA	89	80	86	46	D2	63	0C	EE	E6	7D	A8	:17



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D500:	D5	23	7E	EE	00	CA	20	D5	2B	3E	A5	D3	44	7E	D3	45	:DE
D510:	23	3E	A1	D3	44	7E	D3	45	3E	28	D3	44	3E	F1	D3	45	:73
D520:	23	7E	32	F4	D5	23	7E	32	F5	D5	23	E5	01	00	DA	A7	:C3
D530:	ED	42	E1	CC	3A	D5	22	F6	D5	C9	21	00	D7	C9	00	00	:62
D540:	3E	28	D3	44	3E	02	D3	45	2A	FA	D5	7E	EE	00	C2	59	:55
D550:	D5	23	7E	EE	00	CA	70	D5	2B	3E	A6	D3	44	7E	D3	45	:2F
D560:	23	3E	A2	D3	44	7E	D3	45	3E	28	D3	44	3E	F2	D3	45	:75
D570:	23	7E	32	F8	D5	23	7E	32	F9	D5	23	E5	01	3C	DB	A7	:08
D580:	ED	42	E1	CC	8A	D5	22	FA	D5	C9	21	00	DA	C9	00	00	:B9
D590:	CD	A0	D5	3A	9F	D5	3C	32	9F	D5	C9	00	00	65	D4	00	:D4
D5A0:	3A	9F	D5	EE	08	C0	3E	00	32	9F	D5	3E	07	D3	44	3E	:E2
D5B0:	38	D3	45	2A	9D	D5	3E	08	D3	44	3E	0F	D3	45	3E	01	:ED
D5C0:	D3	44	3E	00	D3	45	D3	44	7E	D3	45	23	E5	01	6F	D4	:66
D5D0:	A7	ED	42	E1	CC	DB	D5	22	9D	D5	C9	21	65	D4	3E	C9	:F1
D5E0:	32	90	D5	3E	07	D3	44	3E	FF	D3	45	C9	00	00	00	00	:11
D5F0:	00	00	00	D6	00	00	00	D7	00	00	00	DA	00	00	00	00	:87
sum :	39	3D	7C	91	1E	DF	ED	82	52	35	7D	AA	C9	FF	C6	97	:C2

## リスト3-3-3 BGMデータ変換プログラム

```

1000 REM ***** THE BGM HENKANN PROGRAM BY KZHS.ISD *****
1010 '
1020 WIDTH 40,25:CONSOLE 0,25,0,0:PRINT CHR$(12);
1030 '
1040 PRINT '***** THE BGM HEN *****'
1050 PRINT '          BY ISHIDA'
1060 PRINT
1070 PRINT '          MENU'
1080 PRINT
1090 PRINT '    1...POKE DATA'
1100 PRINT '    2...MOVE PROG'
1110 PRINT '    3...0A8H→044H'
1120 PRINT '    4...044H→0A8H'
1130 PRINT '    5...END'
1140 PRINT
1150 INPUT '          No(1-5) ' ;NO
1160 ON NO GOSUB 1190,2400,2730,2920,3110
1170 BEEP:PRINT CHR$(12);:WIDTH 40,25:GOTO 1040
1180 END
1190 '
1200 '   THE HENKAN
1210 '   PLAY → MACHINE DATA
1220 '
1230 WIDTH 80,25
1240 CONSOLE 0,25,1,0
1250 COLOR 0,0,0
1260 PRINT CHR$(12);
1270 PRINT '***** THE HENKAN *****'
1280 PRINT
1290 INPUT 'メイン ルーチン ノ セントウ アドレスハ ' ;MA$
1300 MA=VAL(' &H'+MA$)
1310 PRINT
1320 INPUT 'チャンネル ハ (1-3) ' ;CH
1330 IF CH=1 THEN CA=MA+&H3C:CB=MA+&HEB
1340 IF CH=2 THEN CA=MA+&H42:CB=MA+&H13B
1350 IF CH=3 THEN CA=MA+&H48:CB=MA+&H18B
1360 PRINT
1370 INPUT 'データ カギコミ アドレスハ ' ;AD$
1380 AD=VAL(' &H'+AD$)
1390 AL=VAL(' &H'+RIGHT$( '000'+AD$,2))
1400 AH=VAL(' &H'+LEFT$(AD$,2))
1410 AH=AH-(VAL(' &H'+LEFT$(MA$,2))-PEEK(MA+&H57))
1420 POKE CA,AL:POKE CA+1,AH
1430 POKE CB,AL:POKE CB+1,AH

```



```

1440 INPUT "TEMPO =" ; T
1450 OC=-8:NA=&H18*T
1460 PRINT
1470 PRINT "Start!"
1480 RESTORE 5000
1490 READ A1$,A2$,A3$
1500 IF CH=1 THEN A$=A1$
1510 IF CH=2 THEN A$=A2$
1520 IF CH=3 THEN A$=A3$
1530 IF A$="END" THEN 1600
1540 FOR I=1 TO LEN(A$)
1550 MU$=MID$(A$,I,1):IF MU$=" " THEN 1590
1560 DA=INSTR("CcDdEeFfGgAaBbRr<>Ll",MU$)
1570 IF DA=0 THEN 3160 DATA ERROR
1580 ON DA GOSUB 1700,1700,1810,1810,1850,1850,1890,1890,1930,1930,1970,1970,201
0,2010,2050,2050,2130,2150,2170,2170
1590 NEXT:GOTO 1490
1600 PRINT
1610 PRINT "END!":KEY 2,HEX$(AD):BEEP
1620 IF CH=1 THEN CA=MA+&HDD
1630 IF CH=2 THEN CA=MA+&H12D
1640 IF CH=3 THEN CA=MA+&H17D
1650 AL=VAL("&H"+RIGHT$("000"+HEX$(AD),2))
1660 AH=VAL("&H"+LEFT$(HEX$(AD),2))
1670 AH=AH-(VAL("&H"+LEFT$(MA$,2))-PEEK(MA+&H57))
1680 POKE CA,AL:POKE CA+1,AH
1690 RETURN
1700 ' C (ト) sub
1710 IF MID$(A$,I+1,1)="+" THEN C0=&H22:C1=&H8F:I=I+1:GOTO 1740
1720 IF MID$(A$,I+1,1)="-" THEN C0=&H24:C1=&H8D:I=I+1:GOTO 1740
1730 C0=&H22:C1=&H69
1740 POKE AD,C0+OC:POKE AD+1,C1
1750 NA$=RIGHT$("000"+HEX$(NA),4)
1760 NL=VAL("&H"+RIGHT$(NA$,2))
1770 NH=VAL("&H"+LEFT$(NA$,2))
1780 IF MID$(A$,I+1,1)>"9" THEN POKE AD+2,NL:POKE AD+3,NH:AD=AD+4:RETURN
1790 GOSUB 2240 'カカサ カイセキ
1800 RETURN
1810 ' D (レ) sub
1820 IF MID$(A$,I+1,1)="+" THEN C0=&H22:C1=&HDF:I=I+1:GOTO 1740
1830 IF MID$(A$,I+1,1)="-" THEN C0=&H22:C1=&H8A:I=I+1:GOTO 1740
1840 C0=&H22:C1=&H85:GOTO 1740
1850 ' E (ミ) sub
1860 IF MID$(A$,I+1,1)="+" THEN C0=&H23:C1=&H38:I=I+1:GOTO 1740
1870 IF MID$(A$,I+1,1)="-" THEN C0=&H22:C1=&HDF:I=I+1:GOTO 1740
1880 C0=&H23:C1=&H8A:GOTO 1740
1890 ' F (フ) sub
1900 IF MID$(A$,I+1,1)="+" THEN C0=&H23:C1=&H6A:I=I+1:GOTO 1740
1910 IF MID$(A$,I+1,1)="-" THEN C0=&H23:C1=&H8A:I=I+1:GOTO 1740
1920 C0=&H23:C1=&H38:GOTO 1740
1930 ' G (フ) sub
1940 IF MID$(A$,I+1,1)="+" THEN C0=&H23:C1=&HD5:I=I+1:GOTO 1740
1950 IF MID$(A$,I+1,1)="-" THEN C0=&H23:C1=&H6A:I=I+1:GOTO 1740
1960 C0=&H23:C1=&H9D:GOTO 1740
1970 ' A (ラ) sub
1980 IF MID$(A$,I+1,1)="+" THEN C0=&H24:C1=&H4C:I=I+1:GOTO 1740
1990 IF MID$(A$,I+1,1)="-" THEN C0=&H23:C1=&HD5:I=I+1:GOTO 1740
2000 C0=&H24:C1=&HE:GOTO 1740
2010 ' B (リ) sub
2020 IF MID$(A$,I+1,1)="+" THEN C0=&H22:C1=&H69:I=I+1:GOTO 1740
2030 IF MID$(A$,I+1,1)="-" THEN C0=&H24:C1=&H4C:I=I+1:GOTO 1740
2040 C0=&H24:C1=&H8D:GOTO 1740
2050 ' R sub
2060 R$=MID$(A$,I+1,1)
2070 NA$=RIGHT$("000"+HEX$(NA),4)
2080 NL=VAL("&H"+RIGHT$(NA$,2))
2090 NH=VAL("&H"+LEFT$(NA$,2))
2100 IF R$>"9" THEN POKE AD,0:POKE AD+1,0:POKE AD+2,NL:POKE AD+3,NH:AD=AD+4:RETU
RN
2110 GOSUB 2240 'カカサ カイセキ
2120 RETURN
2130 ' < (オクターブ下) sub
2140 OC=OC-8:RETURN
2150 ' > (オクターブ上) sub
2160 OC=OC+8:RETURN
2170 ' L sub

```



```

2180 L$=MID$(A$,I+1,1)
2190 IF L$="2" THEN NA=&H18*T*2
2200 IF L$="4" THEN NA=&H18*T
2210 IF L$="8" THEN NA=&HC*T
2220 IF L$="1" THEN NA=&H6*T:I=I+1
2230 I=I+1:RETURN
2240 ' ナカサ カイセキ
2250 N$=MID$(A$,I+1,1):NN$=MID$(A$,I+2,1)
2260 IF N$="" THEN NG=NA/T:NK=1:GOTO 2350
2270 IF NN$="" THEN WW=1:GOTO 2300
2280 IF N$="." THEN NG=NA/T:NK=1.5:GOTO 2350
2290 IF NN$>"0" AND NN$<"9" THEN WW=2 ELSE WW=1
2300 N$=MID$(A$,I+1,WW)
2310 NG=&H18*4/VAL(N$)
2320 IF WW=2 THEN I=I+1
2330 N$=MID$(A$,I+2,1)
2340 IF N$="." THEN NK=1.5:I=I+1 ELSE NK=1
2350 NA$=RIGHT$("000"+HEX$(NG*T*NK),4)
2360 NL=VAL("&H"+RIGHT$(NA$,2))
2370 NH=VAL("&H"+LEFT$(NA$,2))
2380 POKE AD+2,NL:POKE AD+3,NH
2390 AD=AD+4:I=I+1:RETURN
2400 REM ██████████ MOVE PROG ██████████
2410 '
2420 PRINT CHR$(12);
2430 PRINT "***** MOVE PROGRAM *****"
2440 BEEP
2450 PRINT
2460 INPUT " 7°07°ラムヲヨミマスカ (y/n) ";YN$
2470 IF YN$="n" OR YN$="N" THEN 2520
2480 WIDTH 80,25:FILES
2490 INPUT " file name=";FI$
2500 INPUT " load address=";AD$
2510 AD=VAL("&H"+AD$):BLOAD FI$,AD:GOTO 2530
2520 INPUT " On Memory Address=";AD$
2530 AD=VAL("&H"+AD$):A1=VAL("&H"+LEFT$(AD$,2))
2540 PRINT:AD$=LEFT$(AD$,2)
2550 INPUT " Move address=";MA$
2560 MA=VAL("&H"+LEFT$(MA$,2))
2570 INPUT " Ok(y/n) ";YN$
2580 IF YN$="n" THEN 2420
2590 PRINT:PRINT "Start!":BEEP
2600 RESTORE 2700:WQ=VAL("&H"+AD$)-PEEK(AD+&H2B)
2610 READ A$:IF A$="END" THEN 2680
2620 A=VAL("&H"+A$)
2630 PA=A+AD:IF A$="5A" AND PEEK(PA)=&HD3 THEN 2610
2640 IF A$="192" AND PEEK(PA)=0 THEN 2680
2650 DA=MA-A1+PEEK(PA)+WQ
2660 POKE PA,DA
2670 GOTO 2610
2680 PRINT:PRINT "END!":BEEP
2690 RETURN
2700 DATA 22,2B,2E,34,37,3A,3D,40,43,46,49,4C,57,5A,72,77,7A,7E,82,87,8A,8E,92,9
7,9A,9E,AA,B0,B7,D4,D9,DE,E5,E8,EC
2710 DATA FA,100,107,124,129,12E,135,138,13C,14A,150,157,174,179,17E,185,188,18C
,192,195,199,19E,1A2,1AA,1B5,1CF,1D6,1D9,1DD,1E2
2720 DATA END
2730 REM ██████████ FM I/O 0A8H ▲ FM I/O 044H ██████████
2740 ' FM オンゲン ホート ▲ PC-8801 SR TR FR MR アンカン
2750 PRINT CHR$(12);
2760 PRINT "***** 0A8H ▲ 044H *****"
2770 PRINT
2780 INPUT " ON MEMORY ADDRESS=";AD$
2790 AD=VAL("&H"+AD$)
2800 INPUT " Ok(y/n) ";YN$
2810 IF YN$="n" OR YN$="N" THEN 2750
2820 PRINT:PRINT "Start!":BEEP
2830 FOR I=AD TO AD+512
2840 IF PEEK(I)=&HD3 THEN GOSUB 2860
2850 NEXT:GOTO 2900
2860 I=I+1
2870 IF PEEK(I)=&HA8 THEN POKE I,&H44:RETURN
2880 IF PEEK(I)=&HA9 THEN POKE I,&H45:RETURN
2890 RETURN
2900 PRINT:PRINT "END!":BEEP
2910 RETURN

```



```

2920 REM ██████████ FM I/O 044H ▲ FM I/O 0A8H ██████████
2930 '          PC-8801 SR TR FR MR ▲ FM オンゲ"ン ホート" ンカン
2940 PRINT CHR$(12);
2950 PRINT "***** 044H ▲ 0A8H *****"
2960 PRINT
2970 INPUT "ON MEMORY ADDRESS=";AD$
2980 AD=VAL("&H"+AD$)
2990 INPUT "Ok(y/n)";YN$
3000 IF YN$="n" OR YN$="N" THEN 2940
3010 PRINT:PRINT "Start!";BEEP
3020 FOR I=AD TO AD+512
3030 IF PEEK(I)=&HD3 THEN GOSUB 3050
3040 NEXT:GOTO 3090
3050 I=I+1
3060 IF PEEK(I)=&H44 THEN POKE I,&HA8:RETURN
3070 IF PEEK(I)=&H45 THEN POKE I,&HA9:RETURN
3080 RETURN
3090 PRINT:PRINT "END!";BEEP
3100 RETURN
3110 REM ██████████ E N D ██████████
3120 '
3130 WIDTH 80,25:PRINT CHR$(12);
3140 PRINT "Good by";BEEP
3150 END
3160 REM ██████████ DATA ERROR ██████████
3170 '
3180 BEEP:PRINT:PRINT "ERROR!"
3190 PRINT:PRINT A$:PRINT:PRINT MU$:END
3200 REM ██████████ MUSIC DATA ██████████
3210 RESTORE 5000
3220 READ A$:IF A$="END" THEN END
3230 READ B$,C$
3240 CMD PLAY A$,B$,C$
3250 GOTO 3220
5000 DATA END,END,END

```

ロードしたら、ユーティリティをRUNします。RUNすると、まずメニューを表示して、作業の選択をします(第3-3-1図)。

#### ①POKE DATA

ユーティリティプログラム中の行番号

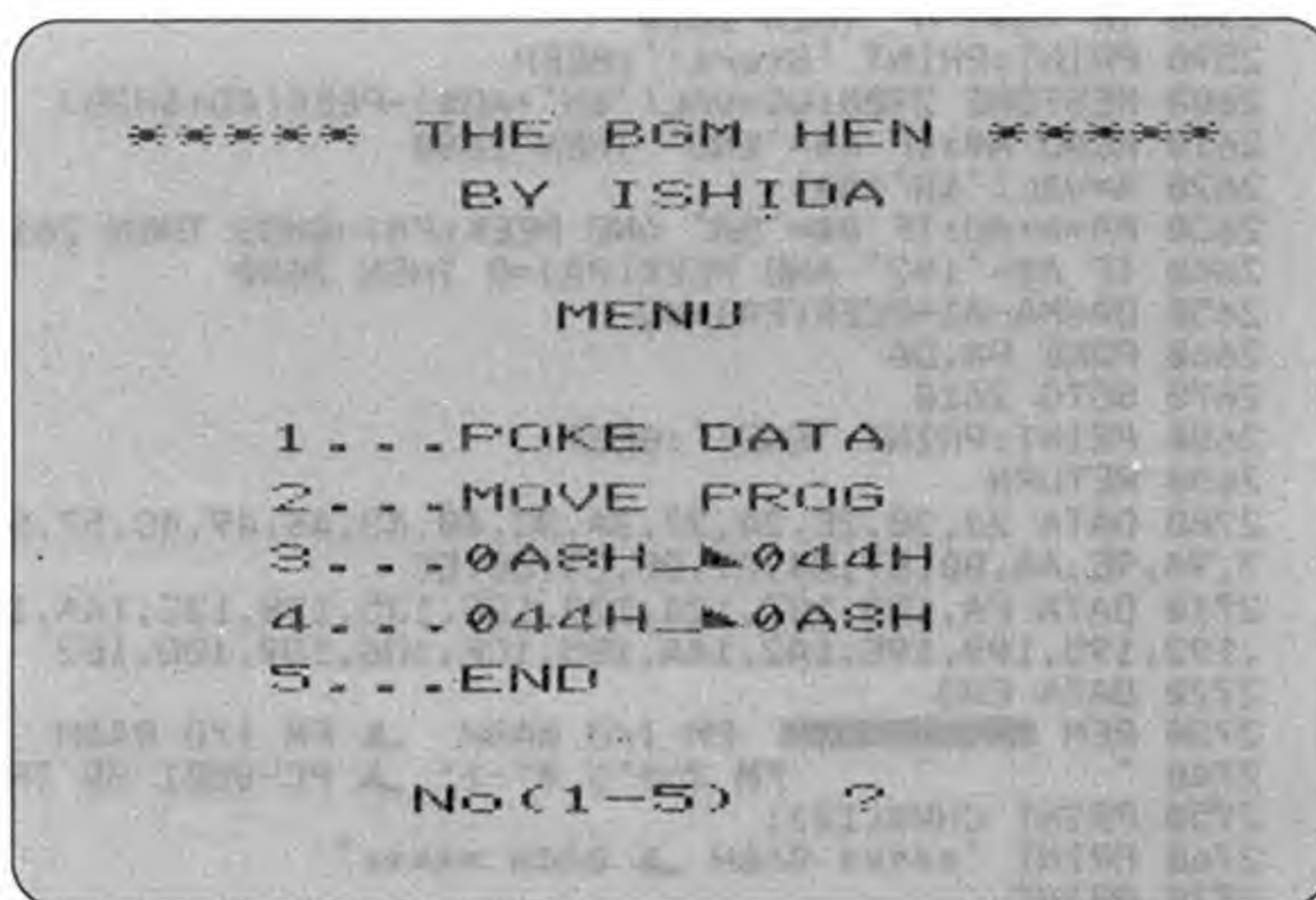
5000

以後に書かれているデータ文中のミュージックデータを第3-2-1表の数値に変換して、メモリに書き込みます。

それでは、5000行以後にミュージックデータを入力してみましょう。ミュージックデータとは、BASICのCMD PLAYで使用する文字列です。つまり、CMD PLAY文のミュージックデータがそのまま使用できます。

しかし、残念ながらPLAY文の文字は、全部使用できません。ここで使える文字は、

C…………ド (アルファベットは、大文字でも小文字でも可)



第3-3-1図 メニュー画面







うにすることです。そうそう、言い忘れましたが、データの最後には必ず“END”(大文字)を三つ入力して下さい。

入力のサンプルとして、GAMMA5のメインBGMのリストをリスト3-3-4に示します。

このリスト3-3-4をユーティリティプログラムの5000番から入力して、POKE DATAを実行すれば、GAMMA5のBGMルーチンができあがります。

POKE DATAは、メニュー画面で、

1<sup>␣</sup>

と入力して下さい。すると画面をクリアして、

メイン ルーチン ノ セントウ アドレスハ?

と聞いてくるので、マシン語のBGMメインルーチンの入っているメモリアドレスを入力して下さい。このとき、アドレスは“&H”はいりません。これからもずっとそうです。又、アドレスは、下位8ビットは00にして下さい。つまり、

C055<sup>␣</sup>や

D3FF<sup>␣</sup>などはできません。

(BGMメインルーチンのロードアドレスも同じです)

BGMメインルーチンを&HD400にロードした人は、

D400<sup>␣</sup>

と入力しましょう。

次に、

“チャンネル ハ (1-3) ?”

と聞いてくるので、データをPOKEするチャンネル番号を入力して下さい。チャンネル1なら、

1<sup>␣</sup>

です。

次に、

“データ カキコミ アドレスハ ?”

とは、そのとおりデータの書き込みアドレスを入力して下さい。&HD600なら

D600<sup>␣</sup>

です。

次に、

“TEMPO = ?”

と聞いてくるので、テンポを入力して下さい。しかし、このテンポは、1が一番速くて、2・3・4……と進むにつれて遅くなります。(小数不可)

普通は、10～16ぐらいです。

とりあえず、

12<sup>␣</sup>

と入力してみましょう。

Start!

と表示した後、しばらくしてメニューにもどります。



もしも、

ERROR!

の表示が出たら、ミュージックデータ中に変換不可能な文字がはいっていますので直して下さい。

変換したデータは、指定されたアドレスに書き込まれています。ファンクションキーの2番には、データのエンドアドレス+1がはいっていますので、次のチャンネルのデータは、そのアドレスから書き込めます。

## ②MOVE PROG

これは、メニューのときに2を入力して下さい。

“プログラムヲ、ヨミコミマスカ (y/n)?”

と聞いてくるので、BGMのメインルーチンを読んでいない人は、

y ☐

と入力して指示に従ってロードして下さい。

次に、

On Memory Address=?

と聞いてくるので、BGMメインルーチンのはいっているアドレスを入力して下さい。

次に、

Move Address?

と聞いてくるので、移動したいアドレスを入力して下さい。OK? (y/n)にyと答えて  
Start!

の表示のあと、しばらくするとメニューにもどります。

Illegal function callのエラーが出た人。あなたはまだデータをPOKEしていませんか?

POKEしてから移動しましょう。

この、MOVE PROGは、モニタのMコマンドのように、単なる

メモリ→メモリ

の移動ではなくて、プログラムの入っているメモリのアドレスは変化しませんが、

プログラム中のアドレスが移動します。

つまり、&HD400用のBGMメインルーチンが、このMOVE PROGを実行しますと、

&HB400用になったり&HC000用になったりします。理解できましたか?

## リスト3-3-5 BGMルーチン&メモリデータ

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D400:	EA	E7	08	E8	50	D4	14	E8	14	E8	14	E8	1A	E8	20	E8	:E3
D410:	F3	3E	2D	D3	44	00	00	00	00	00	00	00	00	00	00	00	:75
D420:	00	3E	D4	ED	47	3E	03	D3	E4	21	50	D4	22	04	D4	21	:9E
D430:	00	00	22	F0	D5	22	F4	D5	22	F8	D5	21	00	D6	22	F2	:CC
D440:	D5	21	00	D7	22	F6	D5	21	00	DA	22	FA	D5	FB	C9	00	:6A
D450:	F3	C5	D5	E5	F5	CD	70	D4	CD	90	D5	3E	03	D3	E4	F1	:93
D460:	E1	D1	C1	FB	C9	EE	77	BD	5E	9F	4F	7E	3F	77	3B	00	:14
D470:	2A	F0	D5	7C	B5	CC	A0	D4	2A	F0	D5	2B	22	F0	D5	00	:61
D480:	2A	F4	D5	7C	B5	CC	F0	D4	2A	F4	D5	2B	22	F4	D5	00	:BD
D490:	2A	F8	D5	7C	B5	CC	40	D5	2A	F8	D5	2B	22	F8	D5	C9	:E3



D4A0:	3E	28	D3	44	3E	00	D3	45	2A	F2	D5	7E	EE	00	C2	B9	:AB
D4B0:	D4	23	7E	EE	00	CA	D0	D4	2B	3E	A4	D3	44	7E	D3	45	:8B
D4C0:	23	3E	A0	D3	44	7E	D3	45	3E	28	D3	44	3E	F0	D3	45	:71
D4D0:	23	7E	32	F0	D5	23	7E	32	F1	D5	23	E5	01	CC	D6	A7	:83
D4E0:	ED	42	E1	CC	EA	D4	22	F2	D5	C9	21	00	D6	C9	00	00	:0C
D4F0:	3E	28	D3	44	3E	01	D3	45	2A	F6	D5	7E	EE	00	C2	09	:00
sum :	87	67	17	C8	2E	89	80	86	46	D2	63	0C	EE	E6	7D	A8	:0A
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D500:	D5	23	7E	EE	00	CA	20	D5	2B	3E	A5	D3	44	7E	D3	45	:DE
D510:	23	3E	A1	D3	44	7E	D3	45	3E	28	D3	44	3E	F1	D3	45	:73
D520:	23	7E	32	F4	D5	23	7E	32	F5	D5	23	E5	01	00	DA	A7	:C3
D530:	ED	42	E1	CC	3A	D5	22	F6	D5	C9	21	00	D7	C9	00	00	:62
D540:	3E	28	D3	44	3E	02	D3	45	2A	FA	D5	7E	EE	00	C2	59	:55
D550:	D5	23	7E	EE	00	CA	70	D5	2B	3E	A6	D3	44	7E	D3	45	:2F
D560:	23	3E	A2	D3	44	7E	D3	45	3E	28	D3	44	3E	F2	D3	45	:75
D570:	23	7E	32	F8	D5	23	7E	32	F9	D5	23	E5	01	3C	DB	A7	:08
D580:	ED	42	E1	CC	8A	D5	22	FA	D5	C9	21	00	DA	C9	00	00	:B9
D590:	C9	A0	D5	3A	9F	D5	3C	32	9F	D5	C9	00	00	65	D4	00	:D0
D5A0:	3A	9F	D5	EE	08	C0	3E	00	32	9F	D5	3E	07	D3	44	3E	:E2
D5B0:	38	D3	45	2A	9D	D5	3E	08	D3	44	3E	0F	D3	45	3E	01	:ED
D5C0:	D3	44	3E	00	D3	45	D3	44	7E	D3	45	23	E5	01	6F	D4	:66
D5D0:	A7	ED	42	E1	CC	DB	D5	22	9D	D5	C9	21	65	D4	3E	C9	:F1
D5E0:	32	90	D5	3E	07	D3	44	3E	FF	D3	45	C9	00	00	00	00	:11
D5F0:	00	00	00	D6	00	00	00	D7	00	00	00	DA	00	00	00	00	:87
sum :	35	3D	7C	91	1E	DF	ED	82	52	35	7D	AA	C9	FF	C6	97	:BE
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D600:	23	0A	18	03	23	0A	84	00	23	38	84	00	22	B5	18	03	:CA
D610:	1C	8D	84	00	22	69	84	00	22	B5	08	01	22	B5	08	01	:FC
D620:	22	B5	84	00	23	9D	08	01	22	B5	84	00	23	0A	18	03	:C7
D630:	1C	0E	84	00	1C	8D	84	00	22	69	8C	01	22	B5	8C	01	:57
D640:	22	69	08	01	1C	8D	08	01	1C	8D	08	01	1C	0E	84	00	:A6
D650:	1B	D5	8C	01	1C	0E	08	01	1C	8D	08	01	22	69	08	01	:F6
D660:	22	B5	08	01	23	0A	10	02	00	00	10	02	23	0A	18	03	:79
D670:	23	0A	84	00	23	38	84	00	22	B5	18	03	1C	8D	84	00	:AF
D680:	22	69	84	00	22	B5	08	01	22	B5	08	01	22	B5	84	00	:2A
D690:	23	9D	08	01	22	B5	84	00	23	0A	18	03	1C	0E	84	00	:1A
D6A0:	1C	8D	84	00	22	69	8C	01	22	B5	8C	01	22	69	08	01	:3D
D6B0:	1C	8D	08	01	1C	8D	08	01	1C	0E	84	00	1B	D5	8C	01	:8F
D6C0:	1C	0E	10	02	00	00	10	02	00	00	20	04	00	00	00	00	:72
D6D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D6E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D6F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	98	85	EC	0A	84	DA	68	0A	66	5C	24	12	81	38	88	0E	:2A
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D700:	1B	0A	84	00	23	0A	42	00	00	00	42	00	1B	0A	84	00	:03
D710:	23	0A	42	00	00	00	42	00	1B	0A	84	00	23	0A	42	00	:C9
D720:	00	00	42	00	1B	0A	84	00	23	0A	42	00	00	00	42	00	:9C
D730:	1A	B5	84	00	22	B5	42	00	00	00	42	00	1A	B5	84	00	:01
D740:	22	B5	42	00	00	00	42	00	1A	B5	84	00	22	B5	42	00	:C7
D750:	00	00	42	00	1A	B5	84	00	22	B5	42	00	00	00	42	00	:F0
D760:	1A	B5	84	00	22	B5	42	00	00	00	42	00	1A	B5	84	00	:01
D770:	22	B5	42	00	00	00	42	00	1A	B5	84	00	22	B5	42	00	:C7
D780:	00	00	42	00	1A	B5	84	00	22	B5	42	00	00	00	42	00	:F0
D790:	1B	0A	84	00	23	0A	42	00	00	00	42	00	1B	0A	84	00	:03
D7A0:	23	0A	42	00	00	00	42	00	1B	0A	84	00	23	0A	42	00	:C9
D7B0:	00	00	42	00	1B	0A	84	00	23	0A	42	00	00	00	42	00	:9C



```

D7C0: 1A 69 84 00 22 69 42 00 00 00 42 00 1A 69 84 00 :1D
D7D0: 22 69 42 00 00 00 42 00 1A 69 84 00 22 69 42 00 :E3
D7E0: 00 00 42 00 1A 69 84 00 22 69 42 00 00 00 42 00 :58
D7F0: 14 8D 84 00 1C 8D 42 00 00 00 42 00 14 8D 84 00 :77
-----
sum : 44 5B AC 00 4C 5B 6A 00 30 CE 6A 00 44 5B AC 00 :0F

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D800: 1C 8D 42 00 00 00 42 00 13 D5 84 00 1B D5 42 00 :CB
D810: 00 00 42 00 13 D5 84 00 1B D5 42 00 00 00 42 00 :22
D820: 14 0E 84 00 1C 0E 42 00 00 00 42 00 14 8D 84 00 :79
D830: 1C 8D 42 00 00 00 42 00 1A 69 84 00 22 69 42 00 :01
D840: 00 00 42 00 1A B5 84 00 22 B5 42 00 00 00 42 00 :F0
D850: 1B 0A 84 00 23 0A 42 00 00 00 42 00 1A B5 84 00 :AD
D860: 22 B5 42 00 00 00 42 00 1A 69 84 00 22 69 42 00 :2F
D870: 00 00 42 00 14 8D 84 00 1C 8D 42 00 00 00 42 00 :94
D880: 1B 0A 84 00 23 0A 42 00 00 00 42 00 1B 0A 84 00 :03
D890: 23 0A 42 00 00 00 42 00 1B 0A 84 00 23 0A 42 00 :C9
D8A0: 00 00 42 00 1B 0A 84 00 23 0A 42 00 00 00 42 00 :9C
D8B0: 1A B5 84 00 22 B5 42 00 00 00 42 00 1A B5 84 00 :01
D8C0: 22 B5 42 00 00 00 42 00 1A B5 84 00 22 B5 42 00 :C7
D8D0: 00 00 42 00 1A B5 84 00 22 B5 42 00 00 00 42 00 :F0
D8E0: 1A B5 84 00 22 B5 42 00 00 00 42 00 1A B5 84 00 :01
D8F0: 22 B5 42 00 00 00 42 00 1A B5 84 00 22 B5 42 00 :C7
-----
sum : 3F CF 6A 00 1C 62 6A 00 34 F1 AC 00 43 D1 6A 00 :AF

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D900: 00 00 42 00 1A B5 84 00 22 B5 42 00 00 00 42 00 :F0
D910: 1B 0A 84 00 23 0A 42 00 00 00 42 00 1B 0A 84 00 :03
D920: 23 0A 42 00 00 00 42 00 1B 0A 84 00 23 0A 42 00 :C9
D930: 00 00 42 00 1B 0A 84 00 23 0A 42 00 00 00 42 00 :9C
D940: 1A 69 84 00 22 69 42 00 00 00 42 00 1A 69 84 00 :1D
D950: 22 69 42 00 00 00 42 00 1A 69 84 00 22 69 42 00 :E3
D960: 00 00 42 00 1A 69 84 00 22 69 42 00 00 00 42 00 :58
D970: 14 8D 84 00 1C 8D 42 00 00 00 42 00 14 8D 84 00 :77
D980: 1C 8D 42 00 00 00 42 00 13 D5 84 00 1B D5 42 00 :CB
D990: 00 00 42 00 13 D5 84 00 1B D5 42 00 00 00 42 00 :22
D9A0: 14 0E 84 00 1C 0E 42 00 00 00 42 00 14 0E 84 00 :FA
D9B0: 1C 0E 42 00 00 00 42 00 14 0E 84 00 1C 0E 42 00 :C0
D9C0: 00 00 42 00 14 0E 84 00 1C 0E 42 00 00 00 42 00 :96
D9D0: 14 0E 84 00 1C 0E 42 00 00 00 42 00 14 0E 84 00 :FA
D9E0: 1C 0E 42 00 00 00 42 00 14 0E 84 00 1C 0E 42 00 :C0
D9F0: 00 00 42 00 14 0E 84 00 1C 0E 42 00 00 00 42 00 :96
-----
sum : 0A 38 6A 00 23 35 AC 00 2A 7D 6A 00 09 80 6A 00 :B4

```

```

      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DA00: 00 00 08 01 2B 0A 08 01 00 00 08 01 2B 0A 08 01 :8E
DA10: 00 00 08 01 2A B5 08 01 00 00 08 01 2A B5 08 01 :E2
DA20: 00 00 08 01 2A B5 08 01 00 00 08 01 2A B5 08 01 :E2
DA30: 00 00 08 01 2B 0A 08 01 00 00 08 01 2B 0A 08 01 :8E
DA40: 00 00 08 01 2A 69 08 01 00 00 08 01 2A 69 08 01 :4A
DA50: 00 00 08 01 24 0E 08 01 00 00 08 01 24 0E 08 01 :88
DA60: 2C 8D 08 01 2B 9D 08 01 2A B5 08 01 24 8D 08 01 :35
DA70: 00 00 84 00 2C 8D 84 00 2B 9D 84 00 2A B5 84 00 :70
DA80: 2C 8D 84 00 2C 8D 84 00 2C 8D 42 00 2B 9D 42 00 :7F
DA90: 2A B5 42 00 00 00 42 00 00 00 08 01 2B 0A 08 01 :AA
DAA0: 00 00 08 01 2B 0A 08 01 00 00 08 01 2A B5 08 01 :38
DAB0: 00 00 08 01 2A B5 08 01 00 00 08 01 2A B5 08 01 :E2
DAC0: 00 00 08 01 2A B5 08 01 00 00 08 01 2B 0A 08 01 :38
DAD0: 00 00 08 01 2B 0A 08 01 00 00 08 01 2A 69 08 01 :EC

```



DAE0:	00	00	08	01	2A	69	08	01	00	00	08	01	24	0E	08	01	:E9
DAF0:	00	00	08	01	24	0E	08	01	24	0E	42	00	24	0E	42	00	:2C
<hr/>																	
sum :	82	CF	B2	0D	73	A1	B2	0D	A5	ED	70	0D	8D	D7	70	0D	:D3
<hr/>																	
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
DB00:	24	0E	42	00	24	0E	42	00	24	0E	42	00	24	0E	42	00	:D0
DB10:	24	0E	42	00	24	0E	42	00	24	0E	42	00	24	0E	42	00	:D0
DB20:	24	0E	42	00	24	0E	42	00	24	0E	42	00	24	0E	42	00	:D0
DB30:	24	0E	42	00	24	0E	42	00	00	00	20	04	00	00	00	00	:0C
DB40:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DB50:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DB60:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DB70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DB80:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DB90:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBA0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBC0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBD0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DBF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
<hr/>																	
sum :	90	38	08	00	90	38	08	00	6C	2A	E6	04	6C	2A	C6	00	:7C

さて、自分の好きな曲をメモリにのせて、好きなアドレスにプログラムを移動したら、いよいよBGMの実行です（サンプルとして、GAMMA5のBGMのメモリデータと、BGMルーチンをリスト3-3-5に示します）。

まずメインBGMのイニシャライズです。これは、BGMメインルーチンの入っているアドレスが、&HD400からなら

```
A=&HD410:CALL A
```

です。入っているアドレスが

&HC000からなら

```
A=&HC010:CALL A
```

です。つまり、BGMメインルーチンのスタートアドレス+&H10を、コールして下さい。

さあ次に、音色設定です。

先程のマシン語のルーチンでもいいし、リスト3-2-2でもいいですから、3チャンネルとも何か音色設定をして下さい。

最後は、リアルタイム割り込みの許可ですが、これには、I/Oポートの&HE6のみを設定して下さい。&HE4ポートの割り込みコントローラーは、BGMのメインルーチン内で設定しているので、いじらなくて結構です。

リアルタイム割り込みの許可は、

```
OUT &HE6,7
```

です。

リターンキーを押したとたんに、BGMが流れ出しましたね？ うまくBGMが流れた人、おめでとうございます。あなたは自分でBGMを鳴らしたのです。このBGMを活用し、いろんな事に役立



て下さい。

リターンのとたんに一発暴走した人。BGMルーチンのアドレスは合っていますか？ コールアドレスを間違えてませんか？ ダンプリストはちゃんと入力しましたか？ もう一度ガンバリましょう。

暴走はしなかったものの、音の出ない人。データは入力しましたか？ 音色設定のデータは間違っていないですか？ ボリュームは小さすぎませんか？ もう一回見直しましょう。

めでたく音の出た人。曲が鳴っているのはいいのですが、何だか3チャンネルそれぞれテンポずれていませんか？ している人もいるでしょう。

その理由は、BASICのPLAY文では、

```
10 CMD PLAY "C4","E2"
```

```
20 CMD PLAY "G8","C4"
```

などのプログラムでは、10行も20行も自分の思ったとおり和音が出ました。しかし、このBGMルーチンでは、今のプログラムを実行するには、次のようにしなければならないのです。

```
10 CMD PLAY "C4R4","E2"
```

```
20 CMD PLAY "G8R8","C4"
```

とこうすれば、BGMルーチンはOKです。

### 3-3-2

## 最後のプレゼント

プレゼントと言っても、プログラムです。

リスト3-3-6(アセンブルリスト)と、

リスト3-3-7(ダンプリスト)です。

このマシン語のプログラムは、GAMMA5の効果音を発生するものです(PSG音源による)。

入力するには、

```
CLEAR ,&HCFFF
```

としてからMONで入力しましょう。

使い方は簡単で、イニシャライズではなくて、

```
A=&HD000:CALL A
```

は敵のバクハツ音

```
A=&HD003:CALL A
```

はBEAMの発射音

```
A=&HD006:CALL A
```

ははじきとばされ音

```
A=&HD009:CALL A
```

は電撃音です。

自作ゲームなどに活用すれば、とても楽しくなるでしょうね。



## リスト3-3-7 PSG効果音プログラム(ダンプリスト)

D000	C3	0C	D0	C3	3D	D0	C3	6E	D0	C3	A7	D0	3E	06	D3	44	テ	ミテ=ミテnミテフミ>	トD			
D010	3E	18	D3	45	3E	08	D3	44	3E	10	D3	45	3E	0B	D3	44	>	トE>	トD>	トE>	トD	
D020	3E	50	D3	45	3E	0C	D3	44	3E	18	D3	45	3E	0D	D3	44	>P	トE>	トD>	トE>	トD	
D030	3E	01	D3	45	3E	07	D3	44	3E	07	D3	45	C9	3E	06	D3	>	トE>	トD>	トEノ>	ト	
D040	44	3E	50	D3	45	3E	08	D3	44	3E	10	D3	45	3E	0B	D3	D>P	トE>	トD>	トE>	ト	
D050	44	3E	10	D3	45	3E	0C	D3	44	3E	03	D3	45	3E	0D	D3	D>	トE>	トD>	トE>	ト	
D060	44	3E	0F	D3	45	3E	07	D3	44	3E	07	D3	45	C9	3E	02	D>	トE>	トD>	トEノ>		
D070	D3	44	3E	0D	D3	45	3E	03	D3	44	3E	01	D3	45	3E	09	トD>N	トE>	トD>	トE>		
D080	D3	44	3E	10	D3	45	3E	0B	D3	44	3E	20	D3	45	3E	0C	トD>	トE>	トD>	トE>		
D090	D3	44	3E	18	D3	45	3E	0D	D3	44	3E	01	D3	45	3E	07	トD>	トE>	トD>	トE>		
D0A0	D3	44	3E	3D	D3	45	C9	C5	F5	0E	80	3E	20	D3	40	06	トD>=トEノナ	トEノ>	トEノ>	トEノ>		
D0B0	80	10	FE	AF	D3	40	06	80	10	FE	0C	20	EE	F1	C1	C9	-	ツ	トEノ	-	ノ	チノ

## リスト3-3-6 PSG効果音プログラム(アセンブルリスト)

```

:
: PSG コカ オン
:
:
:      ORG  0D000H
:
D000 C30CD0      JP  SOUNDA
D003 C33DD0      JP  SOUNDB
D006 C36ED0      JP  SOUNDC
D009 C3A7D0      JP  BEEPA
:
D00C 3E06      SOUNDA:LD  A,6          ;ハ"クハツ オン
D00E D344      OUT  (44H),A
D010 3E18      LD  A,18H
D012 D345      OUT  (45H),A          ;R6=18H
D014 3E08      LD  A,8
D016 D344      OUT  (44H),A
D018 3E10      LD  A,10H
D01A D345      OUT  (45H),A          ;R8=10H
D01C 3E0B      LD  A,0BH
D01E D344      OUT  (44H),A
D020 3E50      LD  A,50H
D022 D345      OUT  (45H),A          ;R11=50H
D024 3E0C      LD  A,0CH
D026 D344      OUT  (44H),A
D028 3E18      LD  A,18H
D02A D345      OUT  (45H),A          ;R12=18H
D02C 3E0D      LD  A,0DH
D02E D344      OUT  (44H),A
D030 3E01      LD  A,1
D032 D345      OUT  (45H),A          ;R13=1
D034 3E07      LD  A,7
D036 D344      OUT  (44H),A
D038 3E07      LD  A,7
D03A D345      OUT  (45H),A          ;R7=7 (ALL NOISE ON)
D03C C9        RET
:
D03D 3E06      SOUNDB:LD  A,6          ;BEAM ハツシ" オン
D03F D344      OUT  (44H),A
D041 3E50      LD  A,50H
D043 D345      OUT  (45H),A          ;R6=50H
D045 3E08      LD  A,8
D047 D344      OUT  (44H),A
D049 3E10      LD  A,10H
D04B D345      OUT  (45H),A          ;R8=10H
D04D 3E0B      LD  A,0BH
D04F D344      OUT  (44H),A
D051 3E10      LD  A,10H
D053 D345      OUT  (45H),A          ;R11=10H
D055 3E0C      LD  A,0CH
D057 D344      OUT  (44H),A
D059 3E03      LD  A,3

```



```

D05B D345      OUT  (45H),A          ;R12=3
D05D 3F0D      LD   A,0DH
D05F D344      OUT  (44H),A
D061 3E0F      LD   A,0FH
D063 D345      OUT  (45H),A          ;R13=0FH
D065 3E07      LD   A,7
D067 D344      OUT  (44H),A
D069 3E07      LD   A,7
D06B D345      OUT  (45H),A          ;R7=7 (ALL NOISE ON)
D06D C9        RET

;
D06E 3E02      SOUND: LD  A,2          ;ホーーン
D070 D344      OUT  (44H),A          ;B チャンネル
D072 3E0D      LD   A,221
D074 D345      OUT  (45H),A          ;R2=221
D076 3E03      LD   A,3
D078 D344      OUT  (44H),A
D07A 3E01      LD   A,1
D07C D345      OUT  (45H),A          ;R3=1
D07E 3E09      LD   A,9
D080 D344      OUT  (44H),A
D082 3E10      LD   A,10H
D084 D345      OUT  (45H),A          ;R9=10H
D086 3E0B      LD   A,0BH
D088 D344      OUT  (44H),A
D08A 3E20      LD   A,20H
D08C D345      OUT  (45H),A          ;R11=20H
D08E 3E0C      LD   A,0CH
D090 D344      OUT  (44H),A
D092 3E18      LD   A,18H
D094 D345      OUT  (45H),A          ;R12=18H
D096 3E0D      LD   A,0DH
D098 D344      OUT  (44H),A
D09A 3E01      LD   A,1
D09C D345      OUT  (45H),A          ;R13=1
D09E 3E07      LD   A,7
D0A0 D344      OUT  (44H),A
D0A2 3E3D      LD   A,3DH
D0A4 D345      OUT  (45H),A          ;R7=3DH (CH.B ON)
D0A6 C9        RET

;
D0A7 C5        BEEPA: PUSH BC          ;スパーク
D0A8 F5        PUSH AF
D0A9 0E80      LD   C,80H
D0AB 3E20      BEEPA1: LD  A,20H
D0AD D340      OUT  (40H),A          ;BEEP ON
D0AF 0680      LD   B,80H
D0B1 10FE      DJNZ $                ;WAIT
D0B3 AF        XOR  A
D0B4 D340      OUT  (40H),A          ;BEEP OFF
D0B6 0680      LD   B,80H
D0B8 10FE      DJNZ $                ;WAIT
D0BA 0C        INC  C
D0BB 20EE      JR   NZ,BEEPA1
D0BD F1        POP  AF
D0BE C1        POP  BC
D0BF C9        RET

;
D0C0           END

```

### 3-3-3 最後に

今ここに、BGMの流れるパソコンが完成しました。あなたの88は、BASICでも、マシン語でもBGMを流すことができるようになりました。

あなたは、88の育ての親です。これからも88をどんどん成長させてあげてください。



部屋の中央に様々なパイプやコードが飛び出た、大きくて複雑な機械がある。

マサシ「とうとう出来たぜ！見てくれよ、これっ」

ジュンコ「何んなのよ、それ？ 変な機械ね」

女はポカンとした顔で見る。

マサシ「いいか、よっく聞けよ。俺はなア、アインシュタインもなしえなかった、統一場の理論を完成しちゃったんだ。わかるか？ つまり、重力場も電磁場も皆一つの体系の上に成り立っているって事で……えっさっぱりわかんない。だからさ、俺達がこうして地球の上に立っていられるのは万有引力があるからだろう？ なっ、その引力ってやつも一種の空間のゆがみなわけ。で、磁石のNとSが引き合うのも空間のゆがみなわけ。なに？ ますますわかんない。だからア、水が高いトコから低いトコへ流れるようにだねえ、空間に坂があると思えばいいの。俺達のまわりの空間てのはさ、あっちこっちに凸凹があり、ぐにやぐにやにゆがんでいてえー、つまりそのおー」

熱が入って来ると彼の目は輝きを増し、オカモトタロウの手つきになった。

ジュンコ「あなた、一度お医者さんにみてもらったら？」彼女は不安になった。

マサシ「いや、俺は冷静です」急に真顔になり、口調もガラッと変った。「結論から先に言った方が早い様ですな。この機械は、空間に大きなゆがみを作り、離れた二つの地点を一つに合わせる事ができるというもののなのです。いかに科学にウトイ君と言えども“ワープ”と言う言葉位聞いた事があるでしょう。そうです。これこそワープを起こさせる機械なのです。これさえあれば、この部屋からどんな遠い世界でも、ハワイだろうが、シベリアだろうが、アンドロメダ星雲だろうが行けるんだ」

ジュンコ「冗談じゃないわ、どうやって帰るのよ」

マサシ「俺の計算では、空間には弾力があるから、しばらくすれば元に戻るはずさ」

ジュンコ「いやよっ、私帰るっ！」

マサシ「いいや、だめだね。ケケケケケ、どうなるかわかんないけどスイッチを入れてみるんだ」彼はパネルのボタンを押した。

ジュンコ「だめ！ やめて——」

機械のあちこちのランプがつき、インジケータの針が動き、ブーンと低い音がした。

それにともなって期待に胸をふくらませて待った。振動しだした。マサシは  
部屋もズスとジュンコはふらふらと床に倒れた。機械の様子次第に  
何かが起こっているような感じだ。

振動がおさまリ、機械のランプが消えた。

マサシ「どうしたの、どうなったのよ」

ジュンコ「わからないね。しかし、元の部屋のままとする事は、実験は失敗らしい」

マサシ「良かった、何もなくて……」

彼らはまだ自体に気がついていない。



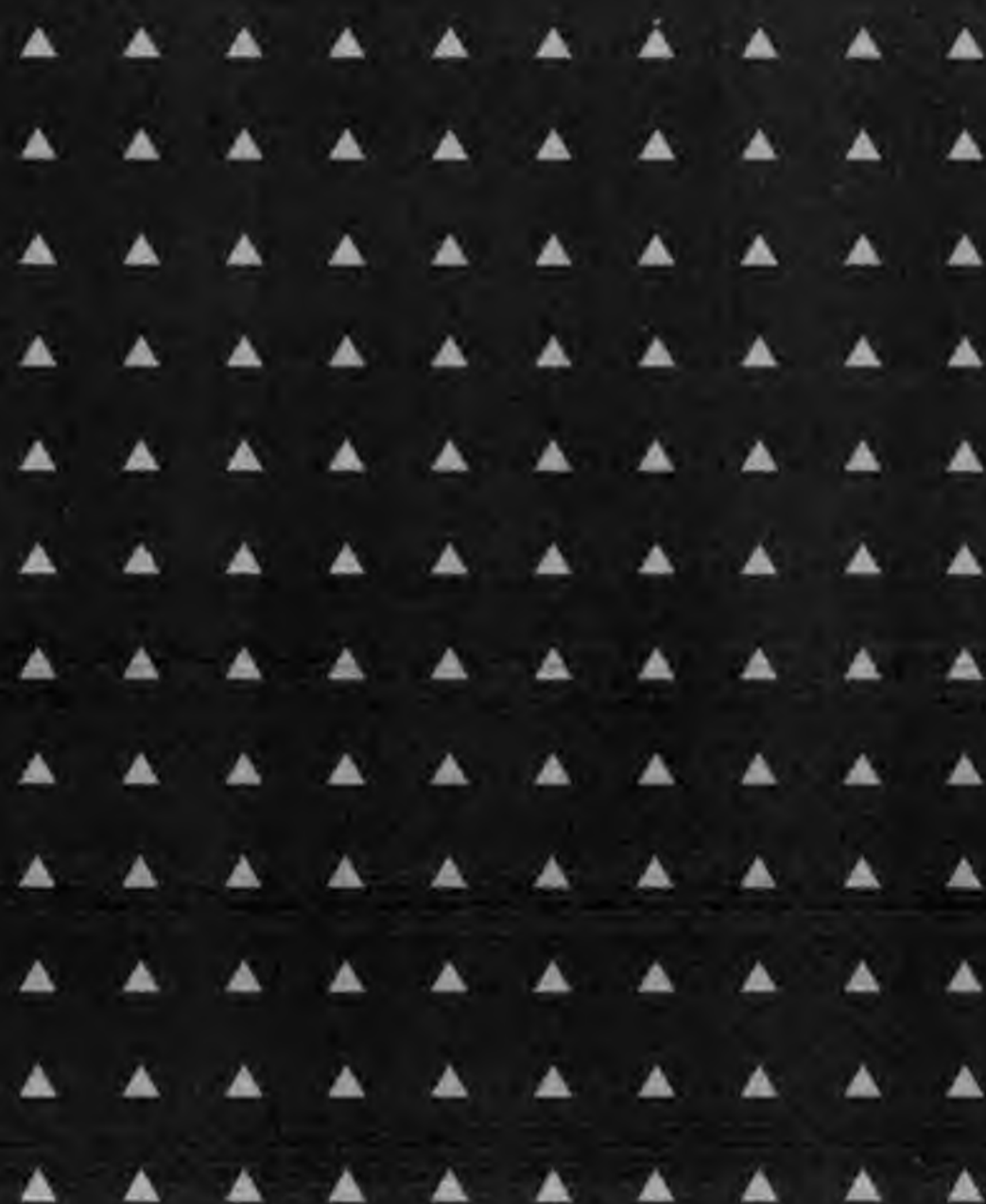
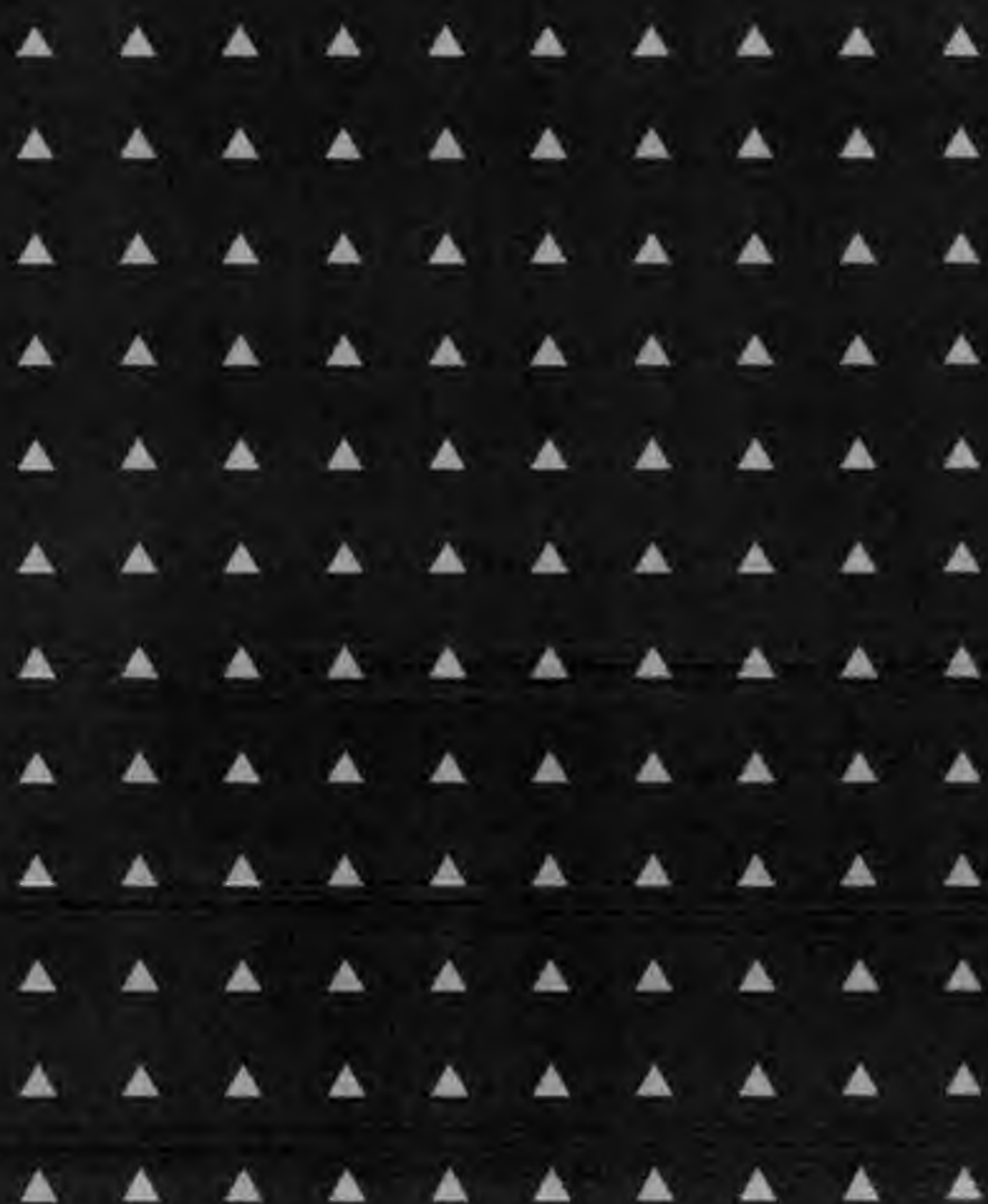
第 4 章

4

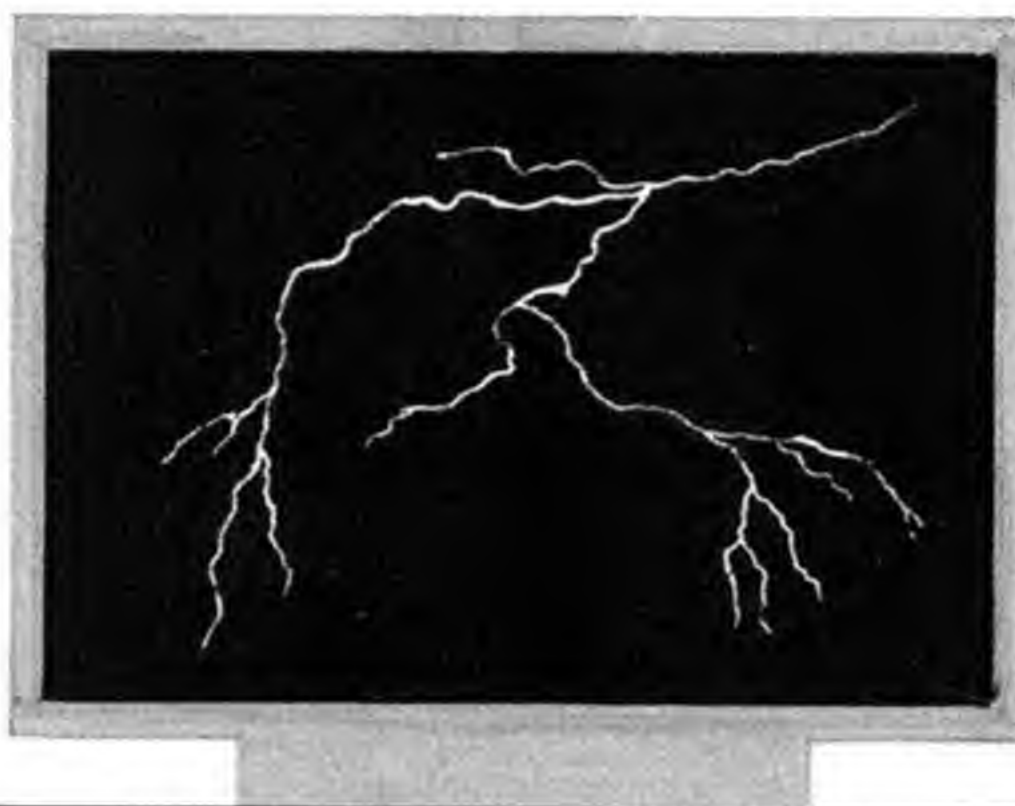
# グラフィック・ハイ・テクニク

4-1 グラフィックデータの圧縮・展開

4-2 高速DISK LOADER







## 4-1 グラフィックデータの圧縮・展開

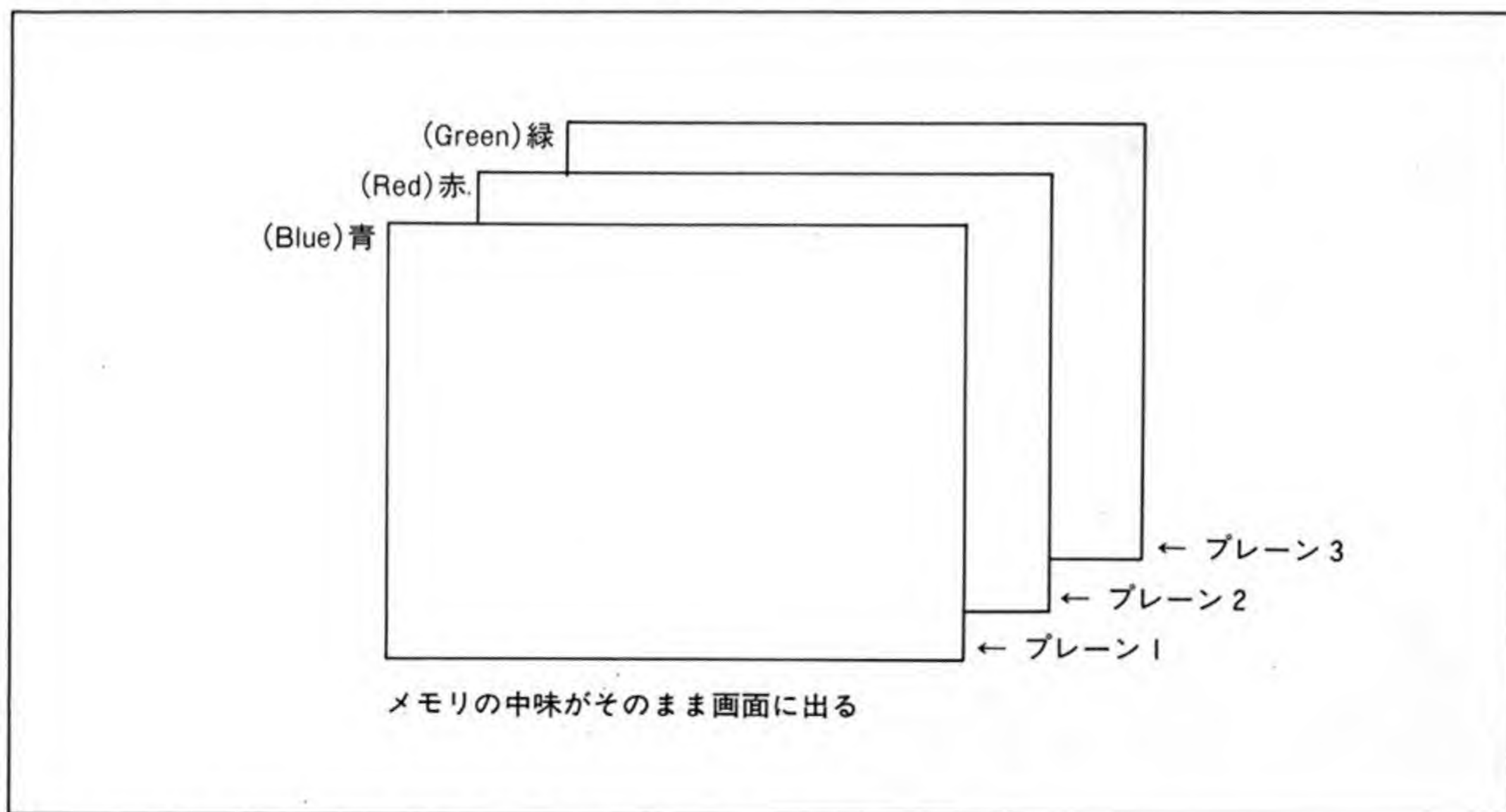
### 4-1-1

#### グラフィックデータの保存方法

最近では8ビット用のグラフィックツールも性能の良い物がいくつも登場し、使いやすさ、スピードともに16ビット用の物に迫るものもあります。

これらのツールはフル画面の絵の情報をディスクにセーブする機能を持っているわけですが、ツールにより、セーブできる枚数がかなり異なります。

その理由は、グラフィックデータの保存方法によります。



第4-1-1図 グラフィック画面の構成(640×200カラーモード)



PC-88では3枚のグラフィック用のプレーンを持ち、横640ドット、縦200ドットの点で絵を表現します。このために必要なメモリは、1ドットを1ビットに対応させて、8ビットで1バイトですから、横は $640 \div 8 = 80$ バイト。これに縦200を掛けて16Kバイト。3プレーンでは $16 \times 3 = 48$ Kバイトになります。

したがって、この情報をそのままディスクに収めようとする、5インチ2D(320K)では何枚も入りません。

もしも、保存する絵が直線で作られた様なものならば、BASICのライン文の始点、終点の座標をデータとしてディスクに収め、絵を書く時にライン文を使って書けばデータはかなり少なくて済みます。

グラフィックツールの中には、この方法で多数の画面データを保存できるものがありますが、描画時間がかかるのと、細かい絵を作るのが難しいという欠点があります。

そこでLINE文やPAINT文によらず、数多くの画面をディスクにセーブするためにはデータの圧縮が必要になってきます。

## 4-1-2

### 圧縮の基本的な考え方

一つの例として、画面のほぼ中央に緑色の点がただ一つある画面を考えてみましょう。

この時、青の画面を担当しているメモリ16Kバイトの中味は全部0になっています。赤も全部0。緑のメモリはC000番地からDF67番地までが0で、DF68番地にあるメモリだけが0以外の数字(たとえば“8”)になっていてそれ以後のメモリは0になっているでしょう。

残念ながら通常はPC-88のモニタでは、直接各プレーンのメモリ内容を見る事はできませんが、もし見る事ができればうんざりする程の00のかたまりを見る事になるでしょう。

さてこの点一つだけの絵をディスクにセーブするとします。全部のデータをそのままセーブするとほとんど0ばかりをセーブする事になりますね。

それではディスクがもったいない。0が16K個ある事を現わすのに0を16K個つめ込むよりは、0のあとに16K個ある事を現す数字を書けば簡単です。

つまり、同じ数のくり返しはくり返しの数(カウンタ)で代用すればデータ量が少なくて済みます。



第4-1-2図 画面の中央に緑色の点がある場合



## 4-1-3

## グラフィックに使われるデータ

データの無い部分、0の所だけを圧縮しても絵によってはあまり影響がありません。

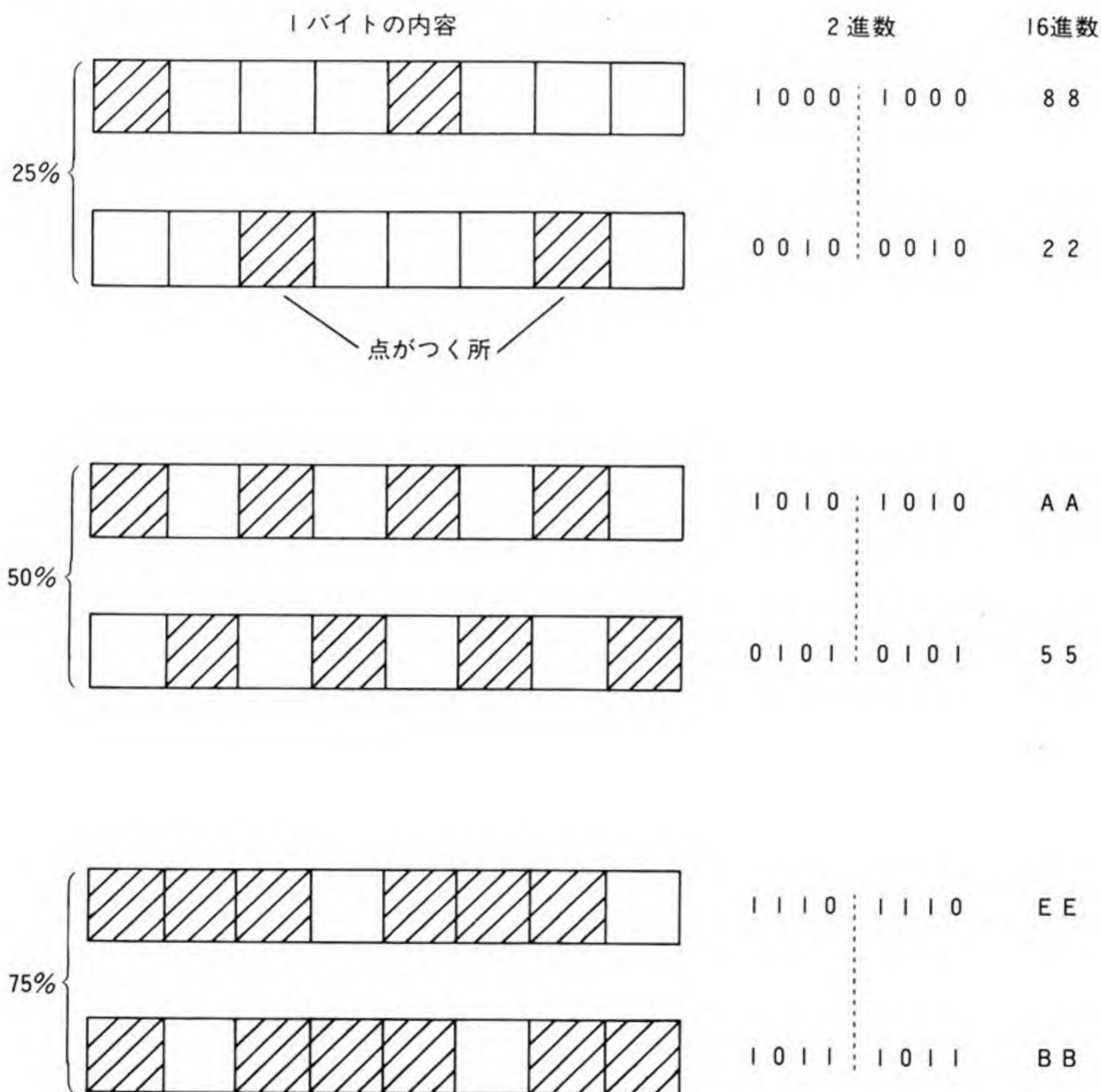
そこで、他にも同じデータが並びそうな所がないかどうかを考えます。

グラフィックツールで書いた絵のデータをモニタで見る事ができるように、うまくメインメモリにのせてみると、"88"や"AA"といったデータの並びが見られます。これは第4-1-3図の様に中間色を表現するため、ある決まった間隔で点を打つとデータも決まった数字になるからです。

これらのデータも圧縮する事を考えましょう。

グラフィックツールでよく使われるパターン

中間色を表現するため上下2バイト一組でアミを作る。



第4-1-3図 中間色の作り方



## 4-1-4

### 圧縮へのいくつかのチャレンジ

グラフィック画面のデータは順番が横に並んでいるので、横に同じ数字を数えていくのが分かりやすく簡単です。

しかし、画面を構成しているメモリは横に80個、縦には200個と、縦に数えて行く方が能率的なようです。その場合には上下二つのデータを一組としてみて行くとよいでしょう。

また、1バイトごとにみてゆくのではなく、ビットごとにカウントする方法などが考えられますが、圧縮の対象となる絵の大きさや細かさ、縦横の比などで最もよい方法を考えます。

## 4-1-5

### 圧縮の具体的な方法

サンプルとして、横1ラインごとに圧縮する方法を考えます。方法は単純ですが、絵によっては少しの改良でさらに圧縮効率を上げる事ができます。

まず、やり方ですが前に述べたようにグラフィックで使われるデータは決まった数です。しかも“55”とか“BB”と言うように、上のケタと下のケタの数字が同じものばかりです。

そこで上と下のケタが同じ数字ならば、次にこのデータがいくつ並んでいるのかカウントする数字を入れます。それ以外のデータならばそのままにしておきます。

展開する時はこの逆をやるわけですが、こうすればカウンタなのかデータなのかプログラムで簡単に見分けがつかますし、プログラムも短いものですみます。

では、プログラムを打ち込んで実際にやってみましょう。

まずBASICのリスト4-1-1をディスクにセーブしておきます。ファイルネームは何でもかまいません。

次にマンション語のリスト4-1-2をモニタのEコマンド等で打ち込み、[BSAVE “GRGEPU”, & HBE00, & H200]として、リスト4-1-1と同じディスクにセーブします。

#### リスト4-1-1 グラフィック圧縮・展開BASIC部

```

100 CLEAR ,&HBDFF:CONSOLE 0,25,0,1
110 IF PEEK(&HBE00)<>&H3E THEN BLOAD "GRGEPU"
120 SCREEN 0,2
130 INPUT "START ADRES HEX NO.= &H";A$
140 A=VAL("&H"+A$)
150 IF A<&HC000 OR A>&HFFFF THEN 130
160 INPUT "X,Y SIZE=";X,Y
170 IF X>80 OR X<1 OR Y>200 OR Y<1 THEN 160
180 POKE &HBFF2,X
190 POKE &HBFF3,Y
200 AH$=LEFT$(A$,2):AL$=RIGHT$(A$,2)

```



```

210 POKE &HBFF0,VAL("&H"+AL$)
220 POKE &HBFF1,VAL("&H"+AH$)
230 GEB=&HBF00:GER=&HBF10:GEG=&HBF20
240 PUB=&HBE00:PUR=&HBE10:PUG=&HBE20
250 INPUT "FILE NAME ?";NA$
260 IF LEN(NA$)>6 THEN 250
270 INPUT "PUT OR GET ? P/G";H$
280 IF H$="P" OR H$="p" THEN 310
290 IF H$="G" OR H$="g" THEN 370
300 GOTO 270
310 BLOAD "2:"+NA$+"B"
320 CALL PUB
330 BLOAD "2:"+NA$+"R"
340 CALL PUR
350 BLOAD "2:"+NA$+"G"
360 CALL PUG:GOTO 430
370 CALL GEB:EAD=PEEK(&HE001):IF EAD>&HE0 THEN 130
380 BSAVE "2:"+NA$+"B",&HC000,(EAD+1-&HC0)*256
390 CALL GER:EAD=PEEK(&HE001):IF EAD>&HE0 THEN 130
400 BSAVE "2:"+NA$+"R",&HC000,(EAD+1-&HC0)*256
410 CALL GEG:EAD=PEEK(&HE001):IF EAD>&HE0 THEN 130
420 BSAVE "2:"+NA$+"G",&HC000,(EAD+1-&HC0)*256
430 SCREEN 0,0:END

```

## リスト4-1-2 グラフィック圧縮展開マシン語部

```

BE00: 3E 5C 32 92 BE 18 1E 00 00 00 00 00 00 00 00 00 :52
BE10: 3E 5D 32 92 BE 18 0E 00 00 00 00 00 00 00 00 00 :43
BE20: 3E 5E 32 92 BE DD 21 00 C0 3A F2 BF 32 F2 BF 3A :E4
BE30: F3 BF 32 F3 BF F3 2A F0 BF 22 90 BE 3A F3 BF 4F :0D
BE40: 3A F2 BF 5F DD 7E 00 DD 23 57 0F 0F 0F 0F E6 0F :2D
BE50: AA E6 0F 28 11 C5 ED 4B 92 BE ED 79 72 D3 5F C1 :F0
BE60: 23 1D 28 1B 18 DE DD 7E 00 DD 23 47 C5 ED 4B 92 :AA
BE70: BE ED 79 72 D3 5F C1 23 1D 28 04 10 EF 18 C5 0D :DE
BE80: 28 0C 2A 90 BE 11 50 00 19 22 90 BE 18 B2 FB C9 :24
BE90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BED0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF00: 3E 5C 32 A6 BF 18 1E 00 00 00 00 00 00 00 00 00 :67
BF10: 3E 5D 32 A6 BF 18 0E 00 00 00 00 00 00 00 00 00 :58
BF20: 3E 5E 32 A6 BF DD 21 00 C0 CD 31 BF DD 22 00 E0 :8D
BF30: C9 F3 2A F0 BF 22 A8 BF 3A F3 BF 4F 3A F2 BF 47 :8B
BF40: C5 C5 ED 4B A6 BF ED 79 7E D3 5F C1 57 0F 0F 0F :82
BF50: 0F E6 0F AA E6 0F 28 1F 7A DD 77 00 23 DD 23 10 :EB
BF60: E0 CD 6A BF C1 0D 20 D4 FB C9 C5 01 50 00 2A A8 :44

```



BF70:	BF	09	22	A8	BF	C1	C9	0E	00	23	0C	C5	ED	4B	A6	BF	:7A
BF80:	ED	79	7E	D3	5F	C1	BA	20	10	10	EE	7A	DD	77	00	DD	:6A
BF90:	23	79	DD	77	00	DD	23	18	C8	2B	7A	DD	77	00	DD	23	:C9
BFA0:	79	DD	77	00	18	B6	00	00	00	00	00	00	00	00	00	00	:9B
BFB0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BFC0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BFD0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BFE0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BFF0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00

グラフィックツールで作成した絵を画面に表示し、BASICに戻します。

ドライブ2にフォーマットしたディスクを用意します。リスト4-1-1をドライブ1からロードし、RUNすると一瞬絵が消えますが、テキストを見やすくするためです。

始めにセーブする画面の左上のアドレスを聞いて来ます。このアドレスは横8ドットずつをxとし、縦をyとすると、 $\&HC000 + 80 \times y + x$ で求められます。

この数字を16進数で答えるのですが、数が分からない時は、このプログラムを走らせる前に、PC-88で〔PRINT HEX\$( $\&HC000 + 80 * \square + \bigcirc$ )〕と、 $\square$ 、 $\bigcirc$ に具体的な数字を入れてリターンキーを押すと、答えが画面に表示されます。左上の隅なら〔C000〕。

なお、xはドットの数ではなくバイトの数、つまり  $0 < x < 80$  の範囲ですので注意して下さい。

次に、xとyのサイズを開いてきますので、xが50、yが80なら、〔50,80〕と打ち込みます。

ファイルネームを聞いてきますから、6文字以内で答えて下さい。

最後に、絵をPUTするのかGETするのか聞いてきます。この場合はGETするのですから〔G〕と押すと、データを青、赤、緑の各プレーンごとに圧縮して、ドライブ2のディスクにセーブします。

圧縮データのファイルネームはB,R,Gの文字がついた三つになります。

この絵をディスクから画面に表示する時は、リスト4-1-1を走らせて、GETした時と同じように各数字を入力し、最後に〔P〕と打ち込めばよいのです。

注意していただきたいのは、圧縮データの領域として、メインRAMの $\&HC000 \sim \&HFFFF$ までの8Kバイトを割りあててありますが、大きな絵や、ブラッシングなどを使った細かい絵はこの中に収まらない事があり、オーバーするとワークエリアに入り込んでBASICが暴走する危険があります。

圧縮データがどこまで入ったかを示す終了アドレスが $\&HE000$ 番地に格納されています。たとえば、E000に“85”があり、E001に“D6”があったとすると、このデータは $\&HD685$ 以前まで使っている事を現わします。

圧縮する時にはyの値を小さくして、モニターで $\&HE001$ の値が“E0”を越えないように確かめながら値を入力するとよいでしょう

またゲームなどに実用化する場合にはディスク用のRAMをバッファに使い、次項で紹介する高速DOSでグラフィックRAMに直接転送するようにします。



## リスト4-1-3 グラフィック圧縮ソースリスト

```

***** SCREEN DATA CONS. PRO. *****
                                by KURE EIJI
                                IN (BFF0)=SCREEN ADDRESS TOP (BFF2)=X,(BFF3),Y
                                ;
                                ;
                                ORG 0BF00H
                                ;
BFF2 SCX: EQU 0BFF2H           ;X SIZE
BFF3 SCY: EQU 0BFF3H           ;Y SIZE
BFF0 GRTOP: EQU 0BFF0H         ;GRAPIC TOP
                                ;
BF00 3E5C           LD A,5CH
BF02 32A6BF         LD (OUTP),A
BF05 181E           JR GRAGET
                                ;
                                ORG 0BF10H
                                ;
BF10 3E5D           LD A,5DH
BF12 32A6BF         LD (OUTP),A
BF15 180E           JR GRAGET
                                ;
                                ORG 0BF20H
                                ;
BF20 3E5E           LD A,5EH
BF22 32A6BF         LD (OUTP),A
                                ;
BF25 DD2100C0 GRAGET: LD IX,0C000H
BF29 CD31BF         CALL MAIN
BF2C DD2200E0       LD (0E000H),IX
BF30 C9             RET
                                ;
BF31 F3             MAIN: DI
BF32 2AF0BF         LD HL,(GRTOP)           ;SCREEN TOP READ
BF35 22A8BF         LD (SCTOP),HL
BF38 3AF3BF         LD A,(SCY)
BF3B 4F             LD C,A                 ;Y SIZE READ
BF3C 3AF2BF         MAIN1: LD A,(SCX)
BF3F 47             LD B,A                 ;X SIZE READ
BF40 C5             PUSH BC
BF41 C5             MAIN2: PUSH BC
BF42 ED4BA6BF       LD BC,(OUTP)
BF46 ED79           OUT (C),A
BF48 7E             LD A,(HL)
BF49 D35F           OUT (5FH),A
BF4B C1             POP BC
BF4C 57             LD D,A                 ;D DATA
BF4D 0F             RRCA
BF4E 0F             RRCA
BF4F 0F             RRCA
BF50 0F             RRCA
BF51 E60F           AND 0FH
BF53 AA             XOR D
BF54 E60F           AND 0FH
BF56 281F           JR Z,COUNT
BF58 7A             LD A,D
BF59 DD7700         LD (IX),A
BF5C 23             MAIN3: INC HL
BF5D DD23           INC IX
BF5F 10E0           DJNZ MAIN2
BF61 CD6ABF         MAIN4: CALL NEXTL
BF64 C1             POP BC
BF65 0D             DEC C
BF66 20D4           JR NZ,MAIN1
BF68 FB             EI
BF69 C9             RET
                                ;
BF6A C5             NEXTL: PUSH BC
BF6B 015000         LD BC,50H
BF6E 2AA8BF         LD HL,(SCTOP)

```



```

BF71 09          ADD HL,BC
BF72 22A8BF      LD (SCTOP),HL
BF75 C1         POP BC
BF76 C9         RET

;
BF77 0E00      COUNT: LD C,0          ;COUNTER=0
BF79 23      COUN1: INC HL          ;IN D DATA
BF7A 0C          INC C
BF7B C5          PUSH BC
BF7C ED4BA6BF   LD BC,(OUTP)
BF80 ED79       OUT (C),A
BF82 7E         LD A,(HL)
BF83 D35F       OUT (5FH),A
BF85 C1         POP BC
BF86 BA         CP D
BF87 2010       JR NZ,COUN2
BF89 10EE       DJNZ COUN1
BF8B 7A         LD A,D          ;SET D
BF8C DD7700     LD (IX),A
BF8F DD23       INC IX
BF91 79         LD A,C
BF92 DD7700     LD (IX),A      ;SET COUNT
BF95 DD23       INC IX
BF97 18C8       JR MAIN4
BF99 2B      COUN2: DEC HL
BF9A 7A         LD A,D          ;SET D
BF9B DD7700     LD (IX),A
BF9E DD23       INC IX
BFA0 79         LD A,C
BFA1 DD7700     LD (IX),A      ;SET COUNT
BFA4 18B6       JR MAIN3

;
;
BFA6 0000      OUTP: DW 0
BFA8 0000      SCTOP: DW 0
;
BFAA          END

```

## リスト4-1-4 グラフィック展開ソースリスト

```

;
; TENKAI PUT
;
BFF2      XSIZ: EQU 0BFF2H
BFF3      YSIZ: EQU 0BFF3H
BFF0      GRTOP: EQU 0BFF0H
;
;      ORG 0BE00H
;
BE00 3E5C      LD A,5CH
BE02 3292BE    LD (OUTP),A
BE05 181E      JR MAIN
;
;      ORG 0BE10H
;
BE10 3E5D      LD A,5DH
BE12 3292BE    LD (OUTP),A
BE15 180E      JR MAIN
;
;      ORG 0BE20H
;
BE20 3E5E      LD A,5EH
BE22 3292BE    LD (OUTP),A
;
BE25 DD2100C0  MAIN: LD IX,0C000H      ;DATA TOP
BE29 3AF2BF    LD A,(0BFF2H)
BE2C 32F2BF    LD (XSIZ),A
BE2F 3AF3BF    LD A,(0BFF3H)
BE32 32F3BF    LD (YSIZ),A

```



```

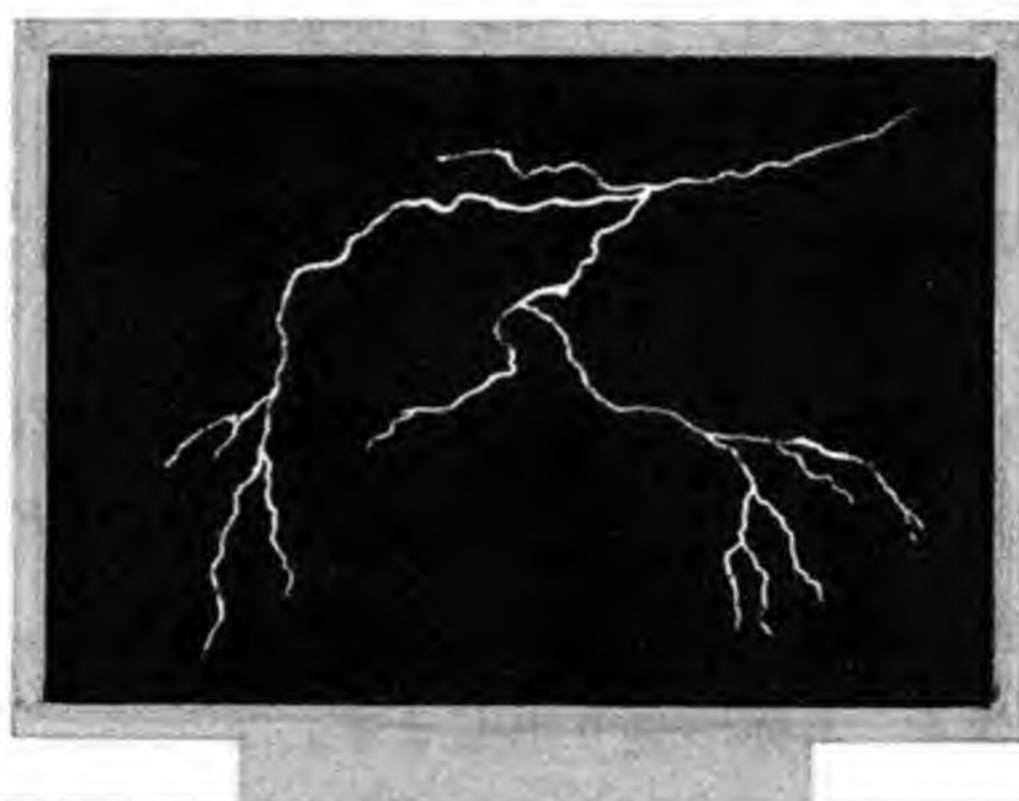
BE35 F3          DI
BE36 2AF0BF      LD  HL,(GRTOP)          ;SCREEN TOP  Y=0,X=0
BE39 2290BE      LD  (SCTOP),HL
BE3C 3AF3BF      LD  A,(YSIZ)
BE3F 4F          LD  C,A                ;Y SIZE
BE40 3AF2BF      MAIN1: LD  A,(XSIZ)
BE43 5F          LD  E,A                ;X SIZE
BE44 DD7E00      MAIN2: LD  A,(IX)
BE47 DD23        INC  IX
BE49 57          LD  D,A                ;D DATA
BE4A 0F          RRCA
BE4B 0F          RRCA
BE4C 0F          RRCA
BE4D 0F          RRCA
BE4E E60F        AND  0FH
BE50 AA          XOR  D
BE51 E60F        AND  0FH
BE53 2811        JR   Z,MAIN3
BE55 C5          PUSH BC
BE56 ED4B92BE    LD  BC,(OUTP)
BE5A ED79        OUT  (C),A
BE5C 72          LD  (HL),D
BE5D D35F        OUT  (5FH),A
BE5F C1          POP  BC
BE60 23          INC  HL
BE61 1D          DEC  E
BE62 281B        JR   Z,MAINEX
BE64 18DE        JR   MAIN2
BE66 DD7E00      MAIN3: LD  A,(IX)          ;COUNT
BE69 DD23        INC  IX
BE6B 47          LD  B,A
BE6C C5          MAIN4: PUSH BC
BE6D ED4B92BE    LD  BC,(OUTP)
BE71 ED79        OUT  (C),A
BE73 72          LD  (HL),D
BE74 D35F        OUT  (5FH),A
BE76 C1          POP  BC
BE77 23          INC  HL
BE78 1D          DEC  E
BE79 2804        JR   Z,MAINEX
BE7B 10EF        DJNZ MAIN4
BE7D 18C5        JR   MAIN2
BE7F 0D          MAINEX: DEC  C
BE80 280C        JR   Z,MAIN5
BE82 2A90BE      LD  HL,(SCTOP)
BE85 115000      LD  DE,50H
BE88 19          ADD  HL,DE
BE89 2290BE      LD  (SCTOP),HL
BE8C 18B2        JR   MAIN1

;
BE8E FB          MAIN5: EI
BE8F C9          RET

;
BE90 0000      SCTOP: DW  0
BE92 0000      OUTP:  DW  0
;
;
BE94          END

```





## 4-2 高速DISK LOADER

### 4-2-1

#### 高速DISK LOADERとは？

この、高速ディスクローダーとは、何でしょう。ディスクローダーとは、フロッピーディスクからCPU本体へプログラム、又はデータを読み込むものです。このローダーは、その転送速度を速くするためのものです。

さあ、それでは一緒に話を進めてみましょう。

### 4-2-2

#### DOSとは何じゃ？

みなさんは、DOSという言葉の意味を知っていますか？ 聞いたことはあると思います。DOS、この3文字は、Disk Operating Systemの略で、頭の3文字を並べたものです。この意味は、ディスクを制御するためのプログラムと考えればよいでしょう。例えばN88-DISK BASICも、DOSの一つです。DISK-BASICを、起動しなければDISKが使えないのは、本体のメモリにDOSが入っていないためです。

### 4-2-3

#### PC-8801のディスクのハード構成

NECのPCシリーズ(8ビット機)の5インチディスクは、すべてインテリジェントタイプと呼ばれています。一体それはなにかというと、要するにディスクユニットの中に、PC本体と同じCPUが入っていて、本体のCPUが直接フロッピーディスクをコントロールしているのではなく、ディスクユ



ニット内のCPUがフロッピーディスクをコントロールしているのです(第4-2-1図)。

そのために、ディスクユニット内に、ちゃんとROMとRAMを持っています(PC-8801mk II・同SR・FR・MR・FH・MHが、CPUを二つ持っているのはこのためです)。

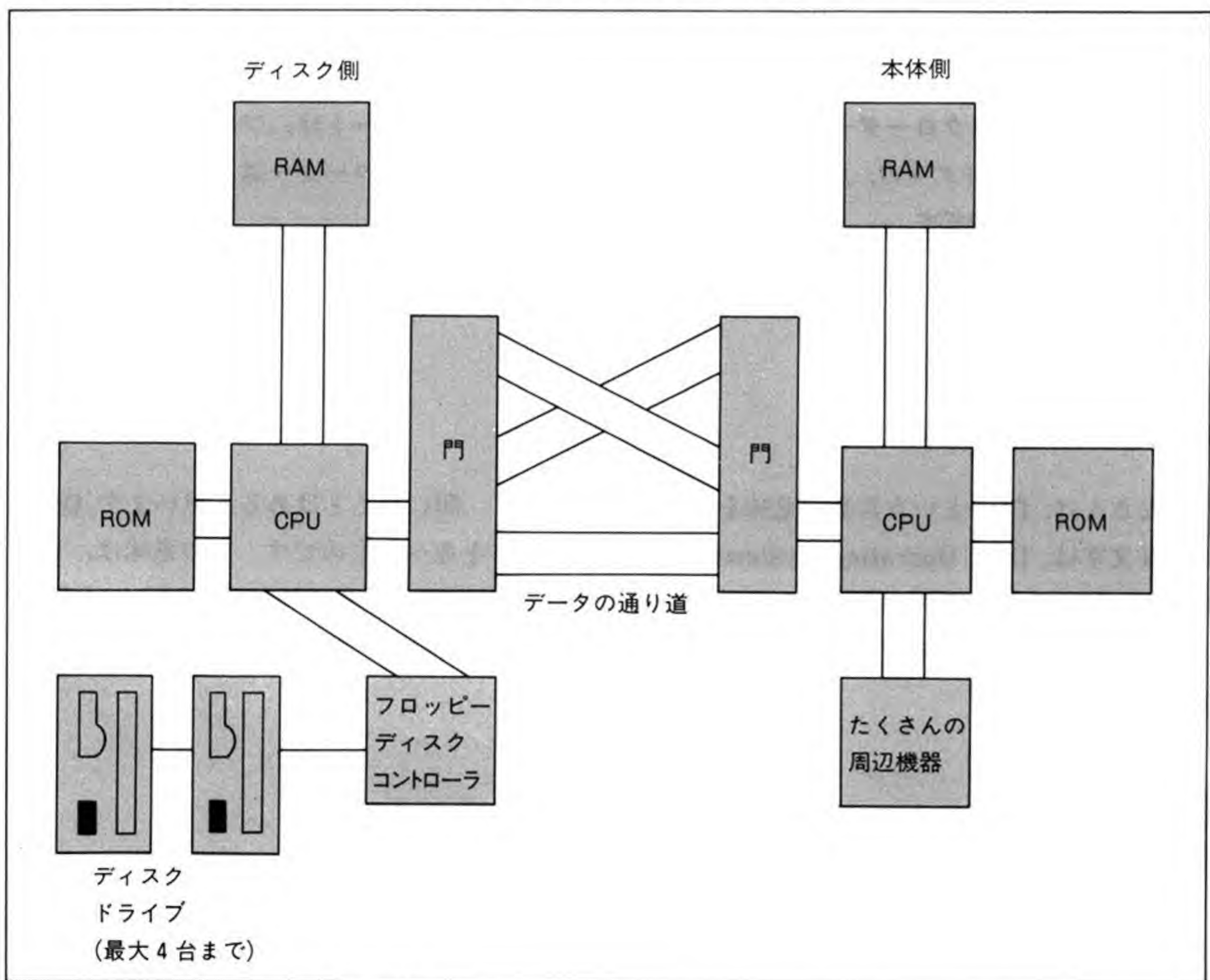
さて、本体のCPUとディスクのCPUとは、どうやって会話(やりとり)をしているのでしょうか?

答えは簡単。すべてコマンドでやりとりしているのです。コマンドとは、ここでは指示ということです。例えば、「ドライブを初期化しろ」とか、「ドライブ1からドライブ2へ、コピーしろ」などです。

本体から、「おい、ドライブ1のフロッピーの、XXの場所を△△だけ読んでくれ」と指示が来ました。するとディスクのCPUは、ドライブ1からデータを自分の(ディスクの)RAMへ読みます。また本体から、「今読んだデータを、全部おくれ」と、指示が来ました。するとディスクのCPUは、自分のRAMの中のさっきのデータを、本体へ送り出します。

と、こんなふうですが、コマンドはもっといろいろあって、「データをディスクに書け」や、「ドライブ2をFORMATしろ」などもあります(本当はまだある)。

さて、PCのディスクのよいところは、このように、ディスクにコマンドを送るだけで、本体は全くドライブをいじる必要がない、というところです。悪いところは、ディスクのRAMから本体のRAMへデータを転送する(またはその逆)のが、遅いというところです。



第4-2-1図 本体とディスクの関係



## 4-2-4

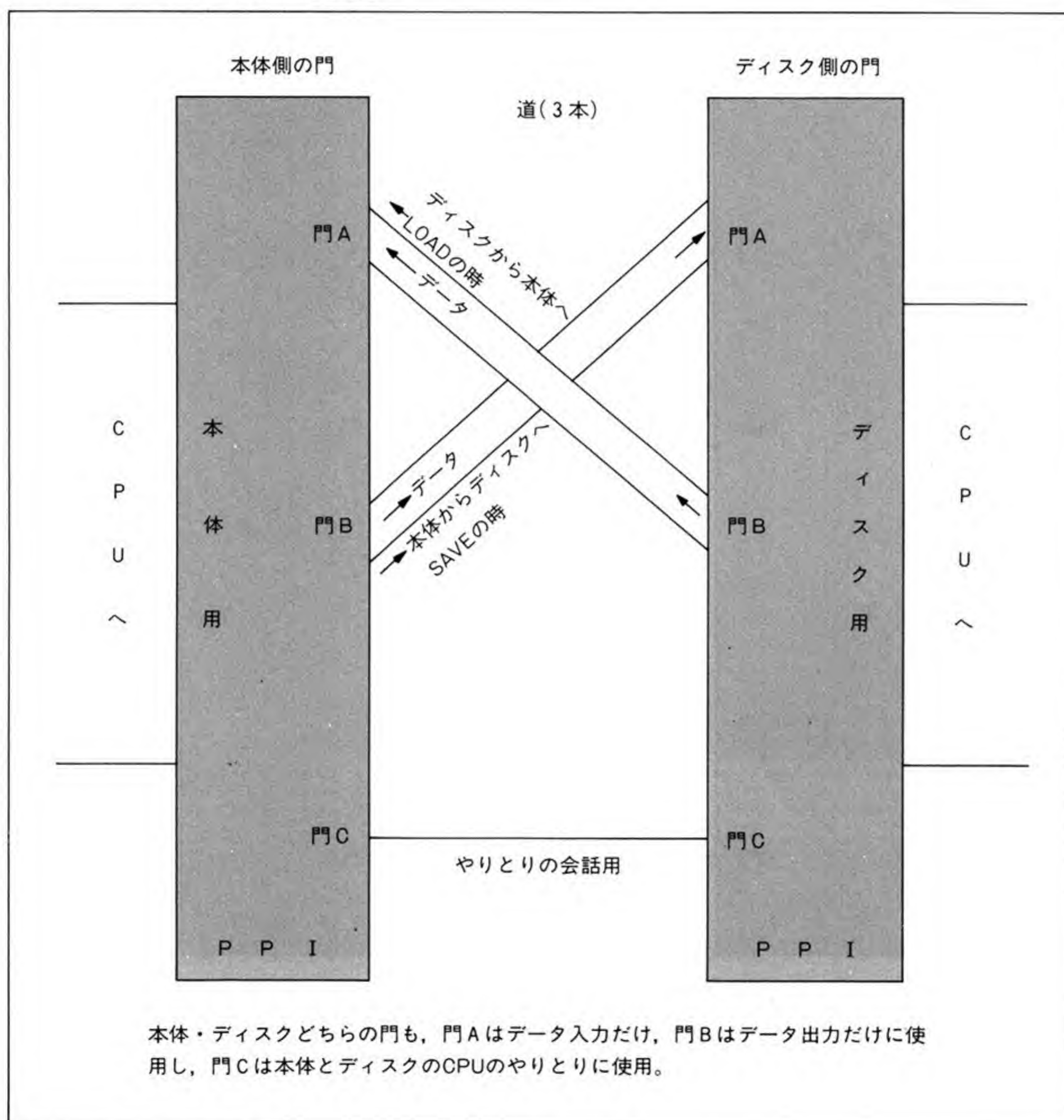
## 何が高速か？

もうおわかりいただけたでしょう。そうです、この高速ディスクローダーは、ディスクのRAMから、本体のRAMへデータを転送するのが高速なのです。

ここでまたちょっとだけ、ハードの話をしなければなりません。

さっき、PCにはメイン(本体の)CPUと、サブ(ディスクの)CPUがあると書きました。この二つのCPUの間の道に、実は“門”があるのです。

この門が、データの行き来を交通整理しています。門とは本当は、PPIと呼ばれるLSIで、道を三つまで整理できます(第4-2-2図)。

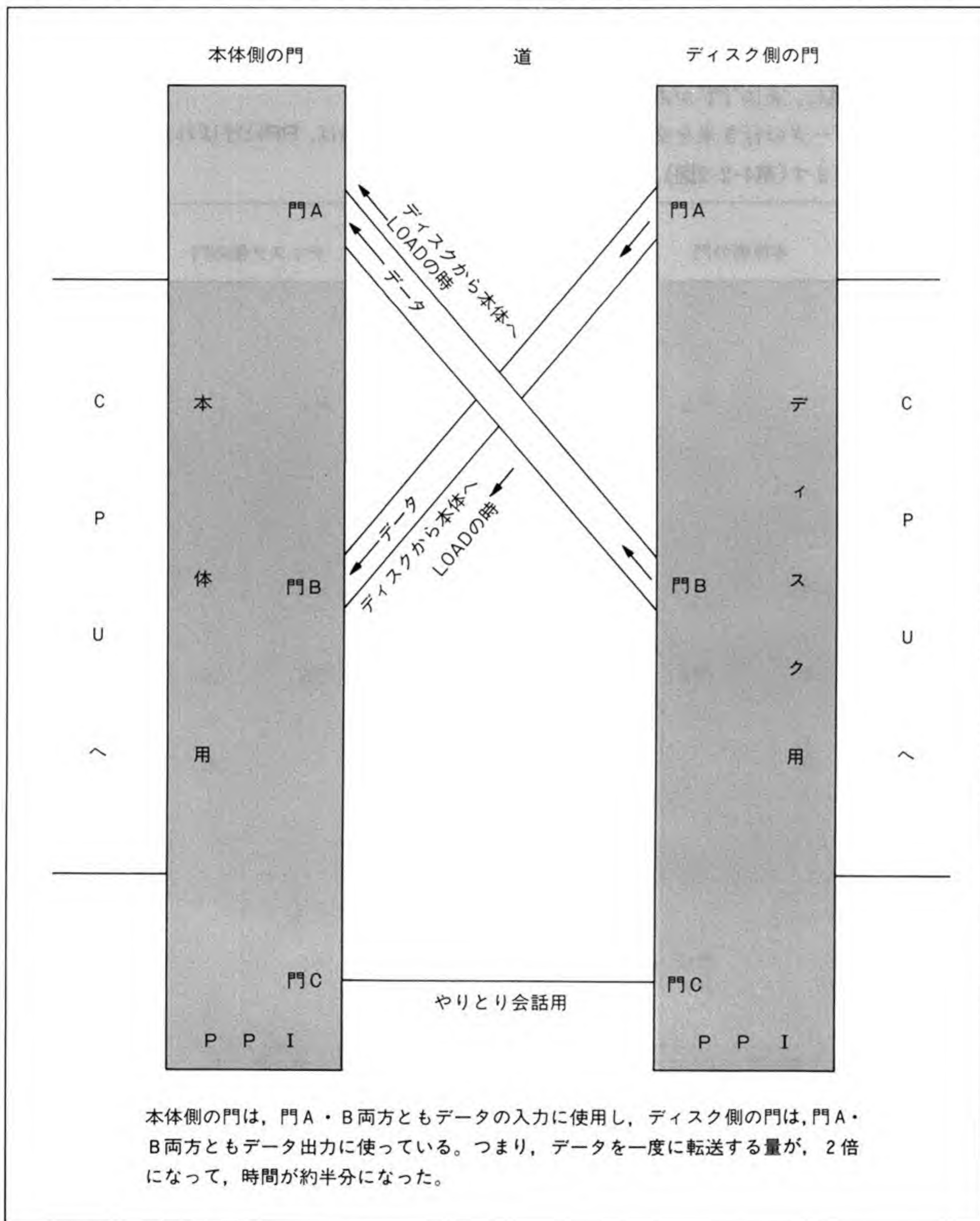


第4-2-2図 普通の門の使い方



PCシリーズでは、このPPIの三つの道すべてを使っています。一つ目は、ディスクのCPUから本体のCPUへのデータの通り道として(LOADのとき)、二つ目は、本体からディスクへの通り道として(SAVEのとき)、そして三つ目は、本体とディスクとのやりとりの会話用に使っています。

普通は、このように使います。ところがところが、「どーせディスクからのロードにしか使わないんだから、セーブのときに使う本体からディスクへの道を、ディスクから本体の方向にして、



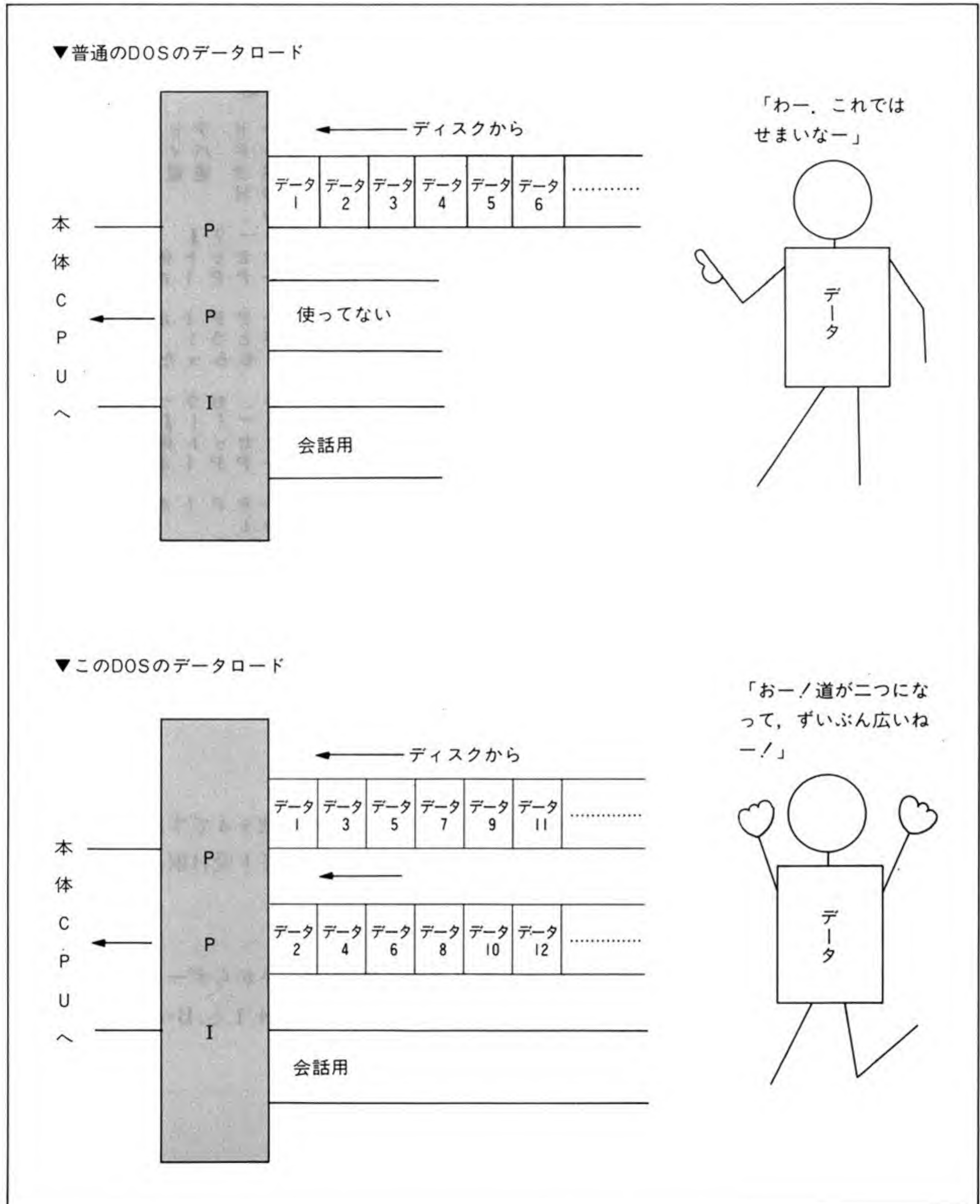
第4-2-3図 このDOSの門の使い方



LOAD用の道を2本にしちゃえ！」と、あるとき思いました。そうですね。LOADとSAVEは一度にできないのだし、どーせあまっているんなら利用しましょうよ(第4-2-3図)。

やっと、高速の理由がわかりました。

今までの話をまとめると、三つある道のうち、今まではそのひとつでデータをもたらっていたのを、二つにしたから速くなった、ということだけです(第4-2-4図)。



第4-2-4図 高速の理由



## 4-2-5 メインルーチンの説明

高速ディスクローダーのメインルーチンを、リストA、リストBに示します。本体用と、ディス

### リストA メインルーチン(本体用)

アドレス	マシンコード	ニーモニック	説明
BE60	21 00 C0	LD HL,0C000H	H L = ロード アドレス
63	11 00 04	LD DE, 0400H	D E = ロード バイト 数 ÷ 4
66	C5	PUSH BC	B C レジスタ 退避
67	0E FD	LD C,0FDH	C ← 0 F D H
69	DB FE	IN A,(0FEH)	『おーい、
6B	E6 01	AND 01H	まだかー?』
6D	28 FA	JR Z,0BE69H	(データセット待ち)
6F	ED A2	INI	(H L) ← P P I ポート B
BE71	0D	DEC C	
72	ED A2	INI	(H L) ← P P I ポート A
74	3E 09	LD A,09H	『ありがとう!
76	D3 FF	OUT (OFFH),A	データもらったよ!』
78	0C	INC C	
79	DB FE	IN A,(0FEH)	『おーい、もう一つ
7B	E6 01	AND 01H	おくれー!!!』
7D	20 FA	JR NZ,0BE79H	(データセット待ち)
7F	ED A2	INI	(H L) ← P P I ポート B
BE81	0D	DEC C	
82	ED A2	INI	(H L) ← P P I ポート A
84	3E 08	LD A,08H	『わーい!
86	D3 FF	OUT (OFFH),A	あんがと!』
88	0C	INC C	
89	1B	DEC DE	D E = D E - 1
8A	7A	LD A,D	
8B	B3	OR E	I F D E < > 0
8C	C2 69 BE	JP NZ,0BE69H	T H E N 0 B E 6 9 H
8F	C1	POP BC	B C レジスタ もどし
BE90	C9	RET	リターン

●HLレジスタがロードアドレスで、DEレジスタはロードバイト数÷4です。なぜ÷4なのかというと、このルーチンは1回のループでデータを4バイト受け取るからです。

●リスト中に、

INI

という命令がありますが、この命令は、「Cレジスタの示すI/Oポートからデータを入力し、HLレジスタの示すアドレスのメモリに書き込み、最後にHL=HL+1と、B=B-1を実行する」というものです。



ク用があります。

説明のかぎかっこの部分が、本体とディスクの会話です。このときに、門Cを使用しています。この会話で、両CPUのタイミングをとっているわけです。

このルーチンだけでは、はっきりいって何もできないので、次のサンプルプログラムへ進みましょう。

# リストB メインルーチン(ディスク用)

ディスク アドレス	マシン コード	アセンブリ	説明
7440	C5	PUSH BC	すべてのレジスタを
41	D5	PUSH DE	退避する。
42	E5	PUSH HL	
43	F5	PUSH AF	
44	21 00 40	LD HL,4000H	HL = 転送データアドレス
47	11 00 04	LD DE,0400H	DE = 転送バイト数 ÷ 4
4A	00	NOP	
4B	00	NOP	
4C	00	NOP	
4D	00	NOP	
4E	00	NOP	
4F	00	NOP	
7450	0E FC	LD C,0FCH	C ← 0FCH
52	ED A3	OUTI	PP Iポート A ← (HL)
54	0C	INC C	
55	ED A3	OUTI	PP Iポート B ← (HL)
57	3E 09	LD A,09H	【データセットしたから、
59	D3 FF	OUT (0FFH),A	読込んでもいいよー。】
5B	0D	DEC C	
5C	DB FE	IN A,(0FEH)	
5E	E6 01	AND 01H	【受け取ったかー？】
7460	28 FA	JR Z,745CH	(本体の、受け取り待ち)
62	ED A3	OUTI	PP Iポート A ← (HL)
64	0C	INC C	
65	ED A3	OUTI	PP Iポート B ← (HL)
67	3E 08	LD A,08H	【よーし、もうひとつ
69	D3 FF	OUT (0FFH),A	セットしたよー。】
6B	0D	DEC C	
6C	DB FE	IN A,(0FEH)	
6E	E6 01	AND 01H	【受け取ってくれたー？】
7470	20 FA	JR NZ,746CH	(本体の、受け取り待ち)
72	1B	DEC DE	DE = DE - 1
73	7A	LD A,D	
74	B3	OR E	IF DE < 0
75	C2 52 74	JP NZ,7452H	THEN 7452H
78	F1	POP AF	
79	E1	POP HL	すべてのレジスタを
7A	D1	POP DE	もどす
7B	C1	POP BC	
7C	C9	RET	りたーん

●このプログラムは、ディスクのRAMにはいります。

●HLレジスタがディスクから読んだデータやプログラムのはいつているアドレスで、DEレジスタは、本体用プログラムと同じで4です。

●OUTIという命令は、INIの逆で、「Cレジスタの示すI/Oポートに、HLの示すアドレスのメモリのデータを出力し、HL=HL+1、B=B-1を実行する」ものです。



## リストC サンプルプログラム

```

100 REM *** High Speed Disk DUMP Program by K.I ***
110 CLEAR ,&HBDFF
120 RESTORE
130 AD=&HBE00:N=340
140 A=0:C=0:FOR I=1 TO 16
150   READ A$
160   IF A$="NEXT" THEN AD=&HBF00:N=N+10:GOTO 140
170   IF A$="END" THEN 260
180   A=VAL("&H"+A$)
190   POKE AD,A:C=C+A
200   AD=AD+1
210 NEXT
220 READ A$:C$=RIGHT$("000"+HEX$(C),2)
230 IF C$=A$ THEN N=N+10:GOTO 140
240 PRINT "LINE";N;"MISS!!"
250 END
260 DEF USR=&HBE00
270 PRINT "All Right!"
280 PRINT "Hit any key to start!!"
290 A$=INPUT$(1)
300 OUT &H51,0
310 I=USR(0)
320 WIDTH 80,25
330 END
340 DATA F3,3E,0C,CD,CC,37,3E,74,CD,D2,37,3E,00,CD,D2,37,A9
350 DATA 3E,01,CD,D2,37,3E,00,CD,D2,37,00,00,00,00,00,29
360 DATA 21,00,BF,06,00,C5,7E,E5,CD,D2,37,E1,23,C1,10,F5,AE
370 DATA 3E,0D,CD,CC,37,3E,74,CD,D2,37,3E,00,CD,D2,37,00,B7
380 DATA 3E,93,D3,FF,06,06,D3,5C,CD,60,BE,D3,5D,CD,60,BE,E4
390 DATA D3,5E,CD,60,BE,10,EF,D3,5F,3E,91,D3,FF,FB,C9,00,B2
400 DATA 21,00,C0,11,00,10,C5,0E,FD,DB,FE,E6,01,28,FA,ED,A1
410 DATA A2,0D,ED,A2,3E,09,D3,FF,0C,DB,FE,E6,01,20,FA,ED,2A
420 DATA A2,0D,ED,A2,3E,08,D3,FF,0C,1B,7A,B3,C2,69,BE,C1,54
430 DATA C9,00,00,00,00,00,00,00,00,00,00,00,00,00,00,C9
440 DATA NEXT
450 DATA 3E,81,D3,FF,00,00,00,00,00,00,00,00,00,00,00,91
460 DATA 16,00,06,24,C5,01,00,10,1E,01,21,00,40,D5,CD,D5,0D
470 DATA 02,D1,14,01,00,10,1E,01,21,00,50,D5,CD,D5,02,D1,D2
480 DATA 14,CD,40,74,C1,10,DD,3E,91,D3,FF,C3,C1,00,00,00,68
490 DATA C5,D5,E5,F5,21,00,40,11,00,08,00,00,00,00,00,EE
500 DATA 0E,FC,ED,A3,0C,ED,A3,3E,09,D3,FF,0D,DB,FE,E6,01,1C
510 DATA 28,FA,ED,A3,0C,ED,A3,3E,08,D3,FF,0D,DB,FE,E6,01,33
520 DATA 20,FA,1B,7A,B3,C2,52,74,F1,E1,D1,C1,C9,00,00,00,17
530 DATA END
540 END

```

**4-2-6****サンプルプログラム&使い方**

このサンプルプログラム(リストC)を入力するときは、N88モードでシステムディスクを立ち



上げた後、そのまま入力して下さい。

入力する時は、行番号はリストどおりに入力し、RENUMは、使用しないで下さい。リストの大部分が、マシン語のデータです。このサンプルプログラムは、チェックサム付きなので、もしもデータに入力ミスがあった場合は、その行番号を表示してくれます。

実行は、RUN<sup>Ⓢ</sup>です。

“LINE×××MISS!!”と、表示されてコマンド待ちになったら、その行番号にバグがあります。もう一度見直してみましょう。

“All Right!”

“Hit any key to start!”

と、表示されたら、ストップキーを押して、プログラムをSAVEしておきましょう。

SAVEし終わったら、実行はRUN 260<sup>Ⓢ</sup>と入力して下さい(すでに、メモリに、データが書き込まれているため)。

さあ、またさっきのメッセージが表示されていますね?

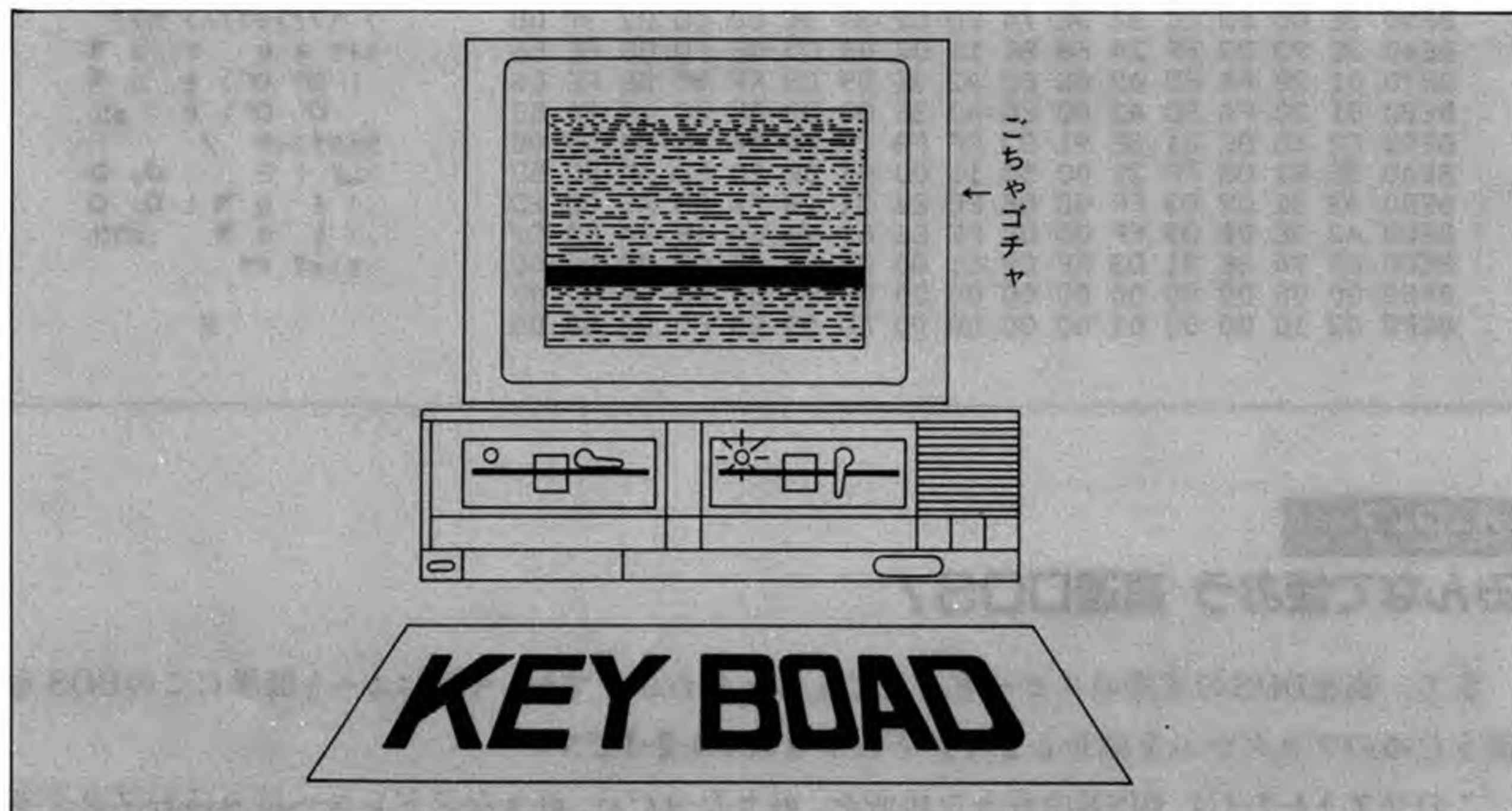
そこで、ドライブ1に何かディスクを入れて、スペースキーを押してみてください。

おやおや? ディスクの赤ランプが光ったとともに、画面にグラフィックのめちゃめちゃしたのが現れてきましたねー! (第4-2-5図)

実は、このサンプルプログラムは、ディスクの中のデータをグラフィック画面に表示する、というものだったのです。

どうですか? このスピード。グラフィックのRAMは、一枚16Kバイトのを、RGBの3枚で合わせて48Kバイトもあるのに、パッパッと8Kバイト単位でどんどん変わっていきます。ディスクのランプも、ほとんど光りっぱなしです(一瞬消えるのは、ドライブのヘッドの移動のため)。

もしも普通のDOSを使ってこんなことを実行したら、48Kバイトのロードなんて、見ていられないほど遅いでしょう。



第4-2-5図 サンプルプログラム実行図



## リスト4-2-1 高速DISK LODERプログラム

```

100 REM *** High Speed Disk Loader ***
110 CLEAR ,&HBOFF
120 DEF USR=&HBE00
130 CLS
140 INPUT 'Drive No.(1-2) ' ;DR
150 IF DR=2 OR DR=1 THEN 160 ELSE 140
160 DR=DR-1
170 INPUT 'Track No.(0-79) ' ;TR
180 IF TR<0 OR TR>79 THEN 170
190 INPUT 'Sector No.(1-16) ' ;SE
200 IF SE<1 OR SE>16 THEN 190
210 INPUT 'Sector ｽｸ (1-16) ' ;SS
220 IF SS<1 OR SS>16 THEN 210
230 INPUT 'Load Address (C000-) ' ;AD$
240 AD$=RIGHT$('000'+AD$,4)
250 AL=VAL('&H'+RIGHT$(AD$,2))
260 AH=VAL('&H'+LEFT$(AD$,2))
270 PRINT
280 INPUT 'Ok (y/n) ' ;YN$
290 IF YN$='N' OR YN$='n' THEN END
300 POKE &HBEF0,2 'Command
310 POKE &HBEF1,SS 'Sector ｽｸ
320 POKE &HBEF2,DR 'Drive No.
330 POKE &HBEF3,TR 'Track No.
340 POKE &HBEF4,SE 'Sector No.
350 POKE &HBEF8,AL 'Address low
360 POKE &HBEF9,AH 'Address High
370 A$=RIGHT$('000'+HEX$(SS*64),4)
380 DL=VAL('&H'+RIGHT$(A$,2))
390 DH=VAL('&H'+LEFT$(A$,2))
400 POKE &HBE68,DL:POKE &HBE69,DH
410 POKE &HBEA8,DL:POKE &HBEA9,DH
420 I=USR(0)
430 END

```

```

BE00 F3 21 F0 BE 7E 23 CD CC 37 7E 23 CD D2 37 7E 23
BE10 CD D2 37 7E 23 CD D2 37 7E 23 CD D2 37 00 00 00
BE20 3E 0C CD CC 37 3E 74 CD D2 37 3E 00 CD D2 37 00
BE30 3E 00 CD D2 37 3E 40 CD D2 37 00 00 00 00 00 00
BE40 21 A0 BE 06 40 7E 23 CD D2 37 10 F9 00 00 00 00
BE50 3E 0D CD CC 37 3E 74 CD D2 37 3E 00 CD D2 37 00
BE60 3E 93 D3 FF 2A F8 BE 11 00 04 C5 0E FD DB FE E6
BE70 01 28 FA ED A2 0D ED A2 3E 09 D3 FF 0C DB FE E6
BE80 01 20 FA ED A2 0D ED A2 3E 08 D3 FF 0C 1B 7A B3
BE90 C2 6D BE C1 3E 91 D3 FF FB C9 00 00 00 00 00 00
BEA0 3E 81 D3 FF 21 00 50 11 00 04 0E FC ED A3 0C ED
BEB0 A3 3E 09 D3 FF 0D DB FE E6 01 28 FA ED A3 0C ED
BEC0 A3 3E 08 D3 FF 0D DB FE E6 01 20 FA 1B 7A B3 C2
BED0 0C 74 3E 91 D3 FF C3 C1 00 00 00 00 00 00 00 00
BEE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BEF0 02 10 00 00 01 00 00 00 00 00 C0 00 00 00 00 00

```

```

月!xセ~#^77~#^x7~#
^x7~#^x7~#^x7
> ^77>t^x7> ^x7
> ^x7>@^x7
! セ @~#^x7
> ^77>t^x7> ^x7
> ト * セ ナ 0 ▼
( O' O' > モ 0 ▼
O' O' > モ zク
ツmセチ>トモ /
> _モ ! P O, O
」> モ 0 ▼ ( O, O
」> モ 0 ▼ zクツ
セ>トモ テチ

```

夕

## 4-2-7

## みんなで使おう 高速DOS!

さて、高速DOSの実際のスピードを見てもらったわけですが、今度はもっと簡単にこのDOSを使うためのプログラムを紹介します。それがリスト4-2-1です。

このリスト4-2-1は、DISKの好きな場所を、好きな分だけ、好きなアドレスへ読み込むプログラムです。入力するときは、始めに、



CLEAR ,&HBDFF ☐

を実行してください。

それから、BASICとマシン語をそれぞれ入力してください(実行前には、必ずSAVE! )。

それでは実行してみましょう。

RUN ☐

とすると、画面がクリアされて、

“Drive No.(1-2) ?”

と表示されます。これは、どのドライブを読みますかと聞いているので、1か2で答えてください。

次に、

“Track No.(0-79) ?”

と表示され、トラック番号を聞いてくるので、0から79で答えてください。

同様に、セクタと、読み込むセクタ数を入力してください。次にロードアドレスですが、これには、なるべく

C000 ☐

としてください(BASICのワークエリアを、壊さないようにするため)。

“OK (y/n) ?”

の問いに、

y ☐

と入力すれば、ディスクのランプが光って、指示されたドライブの指示された場所を、指示されたアドレスに読み込んでいるはずです。第4-2-1表、第4-2-2表に、このプログラムの説明を示します。

いろいろ試してみて、楽しんでください。

第4-2-1表 高速DISK LOADER BASIC部説明

110	マシン語エリアを確保する
120	マシン語プログラムのエントリーアドレス
130	画面クリア
140	ドライブ番号 → DR
170	トラック番号 → TR
190	セクタ番号 → SE
210	読み込みセクタ数 → SS
230	本体のロードアドレス → AD
300	マシン語ルーチンに、ディスクのリードコマンドを書く
310	マシン語ルーチンにセクタ数を書く
320	マシン語ルーチンにドライブ番号を書く
330	マシン語ルーチンにトラック番号を書く
340	マシン語ルーチンにセクタ番号を書く



350, 360	マシン語ルーチンにロードアドレスを書く
370~390	データ転送バイト数÷4を計算
400, 410	データ転送バイト数÷4を, マシン語ルーチンに書く
420	マシン語ルーチンをコールする
430	END

第4-2-1表 高速DISK LOADER マシン語部説明

BE00H~BE1FH	「ディスクの指示された場所をディスクのRAMに読め」と, ディスクのCPUに指示する
BE20H~BE4FH	ディスク用の高速DOSを, ディスクのRAMの7400Hへ転送 する
BE50H~BE5FH	ディスクのCPUを, さっき転送したプログラム(7400H)へ, ジャンプさせる
BE60H~BE9FH	本体用の高速DOSメインルーチン
BEA0H~BEDFH	ディスクへ転送するディスク用の高速DOSプログラム
BEF0H~BEFFH	ワークエリア
※BEF0H .....	ディスクへ送るコマンド (このプログラムは, 02 のディスクリードコマンド)
BEF1H .....	読み込みセクタ数 (1~16)
BEF2H .....	ドライブ番号-1 (ドライブ1は00で, ドライブ 2は01になる)
BEF3H .....	トラック番号 (0~79)
BEF4H .....	セクタ番号 (1~16)
BEF8H .....	本体のロードアドレス下位 1バイト
BEF8H .....	本体のロードアドレス上位 1バイト



第 5 章

5

# Z80マシン語入門

- 5-1 はじめに
- 5-2 コンピュータの成り立ち
- 5-3 マシン語の書き方
- 5-4 レジスタって何だろう？
- 5-5 "LD"命令は基本の基本
- 5-6 "旗(フラグ)"の事
- 5-7 計算させる方法
- 5-8 GOTO(ジャンプ)してみる
- 5-9 GOSUBは知っていますね？
- 5-10 ブロック転送は早いなのなのって
- 5-11 かけ算ができればいいなあ
- 5-12 周辺装置とお話する
- 5-13 その他の命令はあまり気にせず...





## 第5章 Z80マシン語入門

### 5-1 はじめに

「BASICは使えるけどマシン語となると難しくて……」という話しはよく耳にしますが、マシン語が扱いにくいのは間違いありません。

しかし、アクションゲームの様にスピードが要求されるものや、ビジネスプログラムで演算やソート部分にはどうしてもマシン語を使わざるを得ません。

そこで、ここでは誰にでも理解できるように「ゲームのためのZ80マシン語入門」にチャレンジしてみましょう。


マシン語の命令表を一目見ると、沢山のわけのわからぬ文字が並んでいて、始めて見る人には、とても手におえそうな印象を与えます。

でも、ゲーム作りに最低必要な命令となると、さほどの数ではありません。もちろん、沢山の命令を知っていれば、さらによりよくするテクニックも見つけられるでしょうが、それはまた後の話としてよいではありませんか。

とにかく、マシン語を使って何かしてみましょう。慣れれば大した事はないのです。BASICより易しいと思うようになるかもしれません。

“習うより慣れろ”，PC-88にはマシン語を扱うために強力なモニタがついています。


まずはコンピュータのスイッチを入れ、BASICを使えるようにしてから、

**CLEAR, &HBEFF: SCREEN 0, 0: MON** 

と打ち込み、モニタに切り替えましょう。

モニタの使い方はマニュアルに詳しくのっていますが、これからの作業はEコマンドを使います。

モニタでメモリを見たり、メモリに書いたりするのは、このEコマンドが便利です。

h) EBF00 

として、リスト5-1を打ち込んで下さい。間違いなく入力したら、[ESC]キーを押してモニタ




## リスト5-1 マシン語サンプルプログラム

```
BF00 F3 AF D3 51 D3 5E 21 00 80 E5 11 00 C0 01 00 40
BF10 ED B0 E1 11 50 00 A7 ED 52 18 EE FB C9 00 00 00
```

③ 必ずCLEAR, &HBEFFを行ってから実行すること

のコマンド待ちにします。

h] G B F 0 0 

と入力すると、画面に緑の模様が広がり、スクロールを始めます。BASICではとてもこんな高速では画面に書く事ができません。わずか数行のマシン語プログラムでもこんな事ができるのです。

プログラムの内容については後で説明しますが、この様にモニタを使いさえすればマシン語プログラムを作り、それを動かす事ができるのです。あとは紙とエンピツだけでよいのです。

## 5-2

### コンピュータの成り立ち

マシン語は、コンピュータの仕組みに直結した言語です。そこで、コンピュータの仕組みをごくごく簡単に復習してみましょう。

BASICでプログラムを書き、それをコンピュータが実行するとします。

まず、キーボードからプログラムが取り込まれ、それがメモリに乘ります。これを実行させると、コンピュータの中心であるCPU(中央演算処理装置)は、メモリとやりとりしながら各命令を行ってゆき、結果をディスプレイなどに表示します。

このように、コンピュータが仕事をするためには大きく分けて三つの部分が必要です。

- 計算するための頭に相当するCPU
- 仕事の内容や、情報、結果をしまっておくためのメモリ。
- キーボードやディスクなどとコンピュータを結ぶためのインターフェース。

CPUがやりとりをするのは、メモリとインターフェースに対してです。

メモリは仕事の手順や、それに使われるデータが書かれたノートの様なものだと考えます。

それに対してCPUは、内部に計算するために一時的に数値を書き込んでおくメモを持っています。それが“レジスタ”と呼ばれるもので、CPUが計算した値はメモ(レジスタ)に書き込まれます。

では、ある日のCPU君の仕事ぶりを見てみる事にしましょう。③はCPU、④はメモリの内容です。

③「さて、今日も仕事が始まったぞ。では、まずノートの最初の所を見えるか」

④「今日は、15ページの3行目から始めること」

③「えーと、15ページの3行目ね。あったあった」

④「お早よう、CPU。では、さっそくだが、君のメモAに16と書いてもらいたい」



- © 「はいはい、Aに16ね。書きましたよ、次は？」  
M “それに10を足してもらいたい”  
© 「いいですよ、10を足しました。メモAは26に書きかえました」  
M “では、メモAの中味を50ページの1行目に書きなさい”  
© 「はい、お安い御用です。やりましたよ」  
M “それでは休んでよろしい”

以上が16と10を足す時の処理の様子です。CPUに直接16と10を足せとは命令できません。レジスタの中味に対して計算をします。そこで、単純な足し算をするにも、何ステップか命令を与えなければならないのです。

5-3  
マシン語の書き方

リスト5-1を入力したように、マシン語のプログラムは最終的には16進数の00からFFまで256個の数字(コード)の列になります。

しかし、始めからコードを考えていたのでは、とてもプログラムを考えるわけにはいきません。そこで、マシン語もBASICと同じ様に多少人間の言葉(残念ながら英語！)に近い表現で表し、それからコードに直します。

第5-1表はBFF0番地のメモリの中に3と5を足したものを入れるプログラムを比較してみました。BASICでは、

```
POKE &HBFF0, 3 + 5
```

で十分なプログラムですが、マシン語と対応させるために、分解して作ってみました。ていねいに書けばマシン語とBASICは同じように表せます。

表の右側、「ニーモニック」と書いてある所がありますね。このニーモニックがマシン語と人間の理解できる言葉をつなげるものです。

英語の記号で書いてあるので、一見難しいですが、日本語に直せば次のようになります。

```
LD A, 3   →代入しろ  Aに3
ADD A, 5  →足し算    Aと5
LD (0BFF0H), A →代入しろ BFF0番地にA
```

第5-1表 POKE &HBFF0, 3 + 5 プログラム

BASIC	マ シ ン 語	
	マ シ ン コ ー ド	ニ ー モ ニ ッ ク
A = 3	3E 03	LD A, 3
A = A + 5	C6 05	ADD A, 5
POKE &HBFF0, A	32 F0 BF	LD (0BFF0H), A



1行目と3行目はBASICでは違う命令ですが、ニーモニックでは同じ仲間の命令になっていますね。

これからいくつかのニーモニックを覚えなければプログラムは書けませんが、BASICより単純なものばかりです。

3行目に"0BFF0H"とありますが、先頭の0は以下に続く文字がアルファベットではなく、数である事を意味します。最後のHは16進数である事を表します。

もし、アセンブラを使用してコードに変換する場合はこれらの指定が不可欠です。

また、他の人が見て、1かIか混同しないようにするなど、書き方の規則がありますが、ここではハンドアSEMBLなので、自分がわかればよしとして細かい書き方までは考えない事にします。

## 5-4

### レジスタって何だろう？

第5-1表の三つのプログラムにはすべて、"A"という文字が入っています。BASICでは変数と呼ばれますが、マシン語では前に説明したメモに相当するものでレジスタと呼ばれます。

CPUが作業する場合、多くはこのレジスタを通して行います。直接3と5を足しあわせたりする事はできません。

レジスタは、BASICの変数の様なものだと考えてかまいませんが、使える数が決まっていて、それぞれに役割があります。

第5-2表を見て下さい。使えるレジスタとその機能、特徴をまとめてあります。これらのレジスタのすべてを理解したり、使いこなそうとする必要はありません。使えるものから使っていって、だんだんに利用できるようなになればよいのです。

第5-2表 各種レジスタ

#### 8ビットレジスタ

レジスタ	名 称	機 能 ・ 特 徴
A	アキュムレータ (Aレジスタ)	8ビットの算術論理演算はこのレジスタに対して行うものが多く、特にアキュムレータと呼ばれています。
F	フラグレジスタ	このレジスタは、ある計算の状態(0になったりマイナスになったりするなど)を示すフラグで他のレジスタのように変数として使われることはありません
B	Bレジスタ	FOR～NEXTのようなループ作業用に便利な機能を持ちます。
C	Cレジスタ	入出力命令で独自の働きがあります。
D	Dレジスタ	} 一般的な使われ方をします
E	Eレジスタ	
H	Hレジスタ	
L	Lレジスタ	



## 16ビットレジスタ

レジスタ	名 称	機 能 ・ 特 徴
A F	——	16ビットレジスタとして利用はできない。PUSH・POPの時に組にされる。したがってAレジスタを復旧させるとFレジスタも復旧するので注意が必要！
B C	B Cレジスタペア	ブロック転送の時はカウンタとなる
D E	D Eレジスタペア	HLレジスタの補助的性格を持つ
H L	H Lレジスタペア	16ビットの算術論理演算はこのレジスタペアに対して行われるものが多い。  以上は8ビットレジスタを対で使い、16ビットとしたものです。
I X I Y	インデックス レジスタ	HLレジスタのかわりとして使われるほか、相対アドレス指定（オフセットアドレッシング）ができる。ただし、オフセット値の分使うメモリが多くなるので注意。

## 特殊なレジスタ

レジスタ	名 称	機 能 ・ 特 徴
P C	プログラムカウンタ	現在実行中の次のアドレスを示す。一般に利用するものではない。
S P	スタックポインタ	スタックの最下位アドレスを示す。特殊なテクニックでは使う事もある。
I	インタラプトベクタ	割り込みアドレスの指示
R	リフレッシュレジスタ	RAMのリフレッシュ用にたえず変化している。

Z80にはこの他にA～Lまでのレジスタがもう一組(裏レジスタ群)ありますが、一般のA～Lレジスタ(表レジスタ群)と同時に使用する事ができず、本章では扱わない事にします。

第5-2表のように、Z80は16ビットで計算ができます。したがって扱える数字は、0から&HFFFFまで、つまり0から64K(65535)までです。

PC-88のメモリが64K以上あるのは、バンク切り替えと言う手段によるもので、実際にCPUがアクセスできるのは64Kまでです。

16ビットのレジスタの働きで重要なものに、“メモリアドレスを示す”と言うものがあります。

たとえば、&HBFF0にAレジスタの内容を入れたい場合、先程は

```
LD (0BFF0H), A
```

としました。

これと同じ事をHLレジスタを使って行ってみましょう。

```
LD HL, 0BFF0H
```

として、HLレジスタにBFF0を入れました(HレジスタにBFH, LレジスタにF0Hを入れても、もちろん同じ事です)。そして、

```
LD (HL), A
```

と言う命令を実行すればいいのです。

この様に、メモリの中味を示したい時は、カッコをつけます。

では、次からよく使われる命令を見てゆきましょう。



## 5-5

### “LD”命令は基本の基本

まず、今までに何度か登場した“LD”について見てみます。

LDはロードと読みます。そう、プログラムをロードする時のロードですね。意味はもうすでに出て来ました。「代入する」と言う事です。

LD A, C

と書けばAにCの中味を入れる事を表します。

代入は、レジスタからレジスタへの代入、レジスタからメモリへの代入、レジスタへの代入、レジスタからメモリへの代入、レジスタへ数値の代入などなど様々な場合があります。

いくつかの例を上げてみましょう。

LD H, L

⇨ HレジスタにLレジスタの内容を入れる。Lレジスタは変化しない。

LD B, 10H

⇨ Bレジスタに、16進数の10を代入する。

LD A, (0C000H)

⇨ Aレジスタに、C000番地のメモリの内容を入れる。メモリの内容は変化しない。

LD (HL), C

⇨ HLレジスタで示される番地のメモリに、Cレジスタの内容を入れる。

以上の様に、始めのもの(第1オペランド)に、次の内容(第2オペランド)を代入する命令です。第2オペランドの内容は変化することはありません。

他にどんな場合があるのでしょうか。付録Cを見て下さい。それぞれどんな相手とロードする事ができるかがわかります。

縦にあるのが、代入されるもの(第1オペランド)、横にあるのが代入するもの(第2オペランド)で、ますの中にある数字が最終的なマシン語プログラムにするためのコードです。

この16進数のコードが埋っている所の組はロード命令が使える、空白のものはロードできません。たとえば、

LD B, (0D000H)

のようにメモリの内容を直接Bレジスタに代入することはできません。

LD A, (0D000H)

LD B, A

と2段階に分けて、ロードする必要があります。

また、表には

LD HL, A

と言う様な表現はありませんね。言うまでもなく、HLは2バイトの数です。(HL)はHL番地のメモリの中味、メモリは1バイトです。お互いのバイト数があわなくてはロードできないわけです。

では、ロード命令を実際に走らせられるように、マシンコードに直してみましょう。



プログラムを書き、それを16進数のマシン語になおすために、コーディングシートと言うものが市販されていますが、レポート用紙があれば十分でしょう。

まずノートの右、先頭の所にプログラムを入れてゆくメモリの番地を書き、間をあけて、プログラムを書きます。

ニーモニック、マシン語対照表を見ながら中央にマシンコードを書き込んでゆきます(第5-1図)。

モニタを使ってテストする時には、決めた番地に順々にコードを書き込んでゆきます。

この時、重要な事は2バイトの数値を記入する場合、下のケタを先に、上のケタを次に書くと言う事です。第5-1図の3行目で、おやっと思われた方もあるかもしれません。これについては2バイトのロード命令でまた説明します。

テストでGコマンドを使い実行する時には、プログラムの最後に“C9”を書いておくといいでしよう。これは、リターンの意味ですが、モニタはエラーとして処理し、実行後モニタかBASICに戻れます。

では、ロードの様子を画面で見してみましょう。

リスト5-2を打ち込み、h]GBF00

として下さい。わからない命令は気にしないで結構です。途中でタイマーが入れてあるので、ゆっくり画面上方からアミがかかって行くと思います。

ここで、BF0A番地の“55”と言う数字を別な数字、たとえば、03などに置きかえると、たて縞の模様がでてきます。この数字を、

LD (HL), A

で、メモリに代入しているのです。そのメモリはグラフィックRAMで、入れたデータのビットパターンが、そのまま画面に出るわけです。

次に、Z80が、レジスタをペアで使い、2バイト(16ビット)として扱う事は前に書きました。

● 3 + 5 の代入 ← (何のプログラムかわかるようにメモしておく)

アドレス	オブジェクト	ニーモニック
BF00	3E 03	LD A, 3
↑	➤	(数字は16進数でケタをはっきりさせる)
(コーディングの時に決めてもよい)	C6 05	ADD A, 5
	32 F0BF	LD (0BFF0H), A
	↑	(バイトが逆ノ)
	(変換表を見ながらコードに直す)	(作りたいプログラムを書く)

第5-1図 マシン語プログラムの書き方



## リスト5-2a LD命令テストプログラム

```

;
; LD メイレイ テスト
;
;
;          ORG  0BF00H
;
BF00 F3    GPAT: DI                ;ワリコミ キンシ
;
BF01 D35E          OUT  (5EH),A    ;GREEN GRAM
;
BF03 2100C0        LD   HL,0C000H  ;G RAM ADDRESS TOP
BF06 110040        LD   DE,4000H   ;LD スル ナカッサ
;
BF09 3E55    LOOP: LD   A,55H      ;SET BIT PATTERN DATA
BF0B 77          LD   (HL),A       ;SET DATA TO MEMORY
;
BF0C 06FF          LD   B,0FFH
BF0E 10FE    WAIT: DJNZ WAIT      ;WAIT TIMMER
;
BF10 23          INC   HL          ;HL=HL+1
BF11 1B          DEC   DE          ;DE=DE-1
BF12 7B          LD   A,E
BF13 B2          OR    D
BF14 20F3        JR    NZ,LOOP     ;IF DE<>0 THEN LOOP
;
BF16 D35F          OUT  (5FH),A    ;RAM
;
BF18 FB          EI                ;ワリコミ キョカ
;
BF19 C9          RET               ;フブレーク ホﾟイント ノ カワリ
;
;
BF1A          END

```

## リスト5-2b 実行プログラム

```

BF00 F3 D3 5E 21 00 C0 11 00 40 3E 55 77 06 FF 10 FE
BF10 23 1B 7B B2 20 F3 D3 5F FB C9 00 00 00 00 00 00

```

では、この時のロード命令はどうなるでしょう。

付録C表に16ビットのロード命令がまとめてあります。表中のnnとは具体的な16進の数を入れろ、という意味です。

たとえば、

LD HL, 0BFF0H

は、

21 F0 BF

となります。くどいようですが、2バイトの上下を“さかさま”にします。

これを8ビットロード命令にすると、

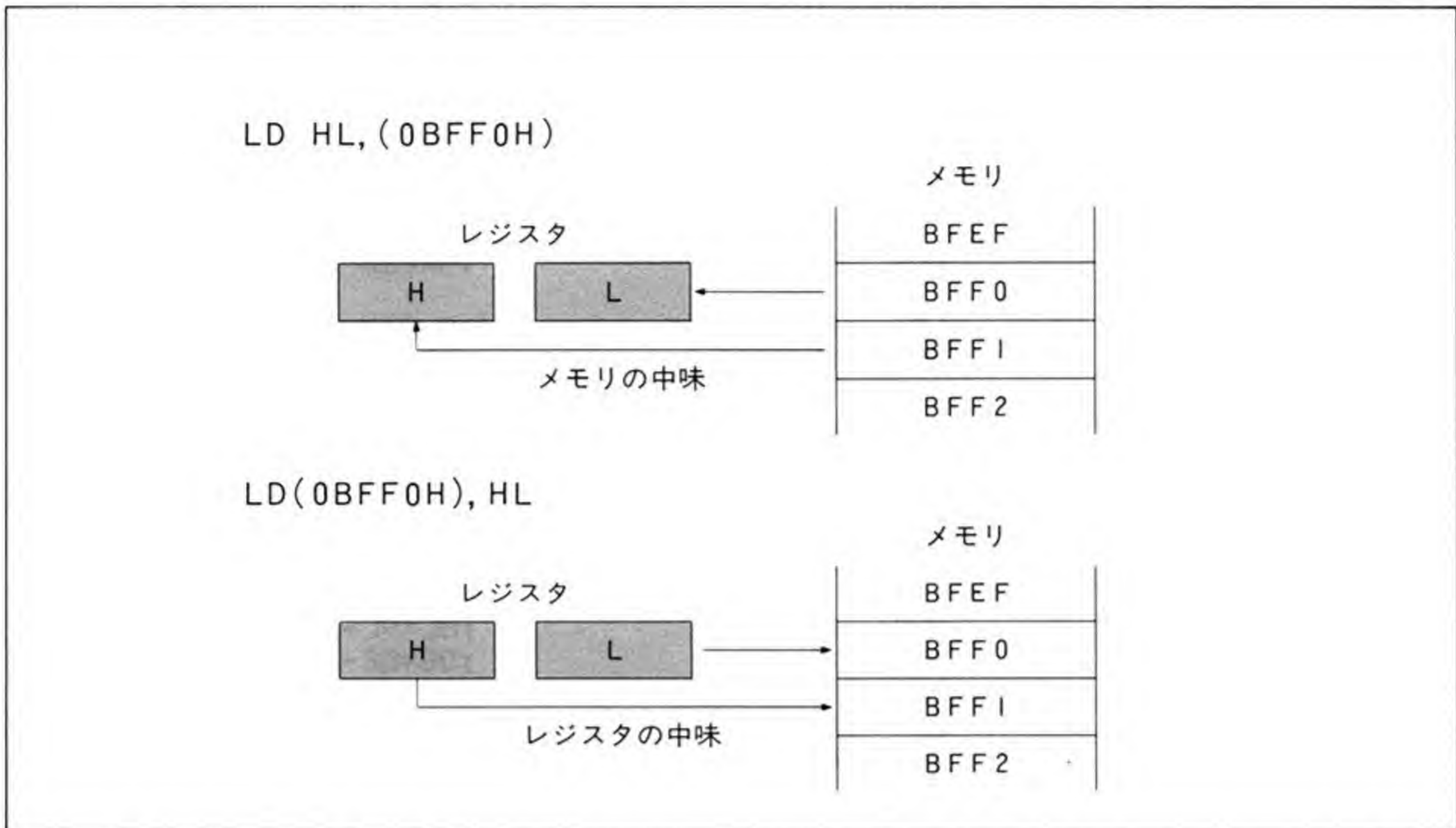
26 BF ← LD H, 0BFH

2E FD ← LD L, 0F0H

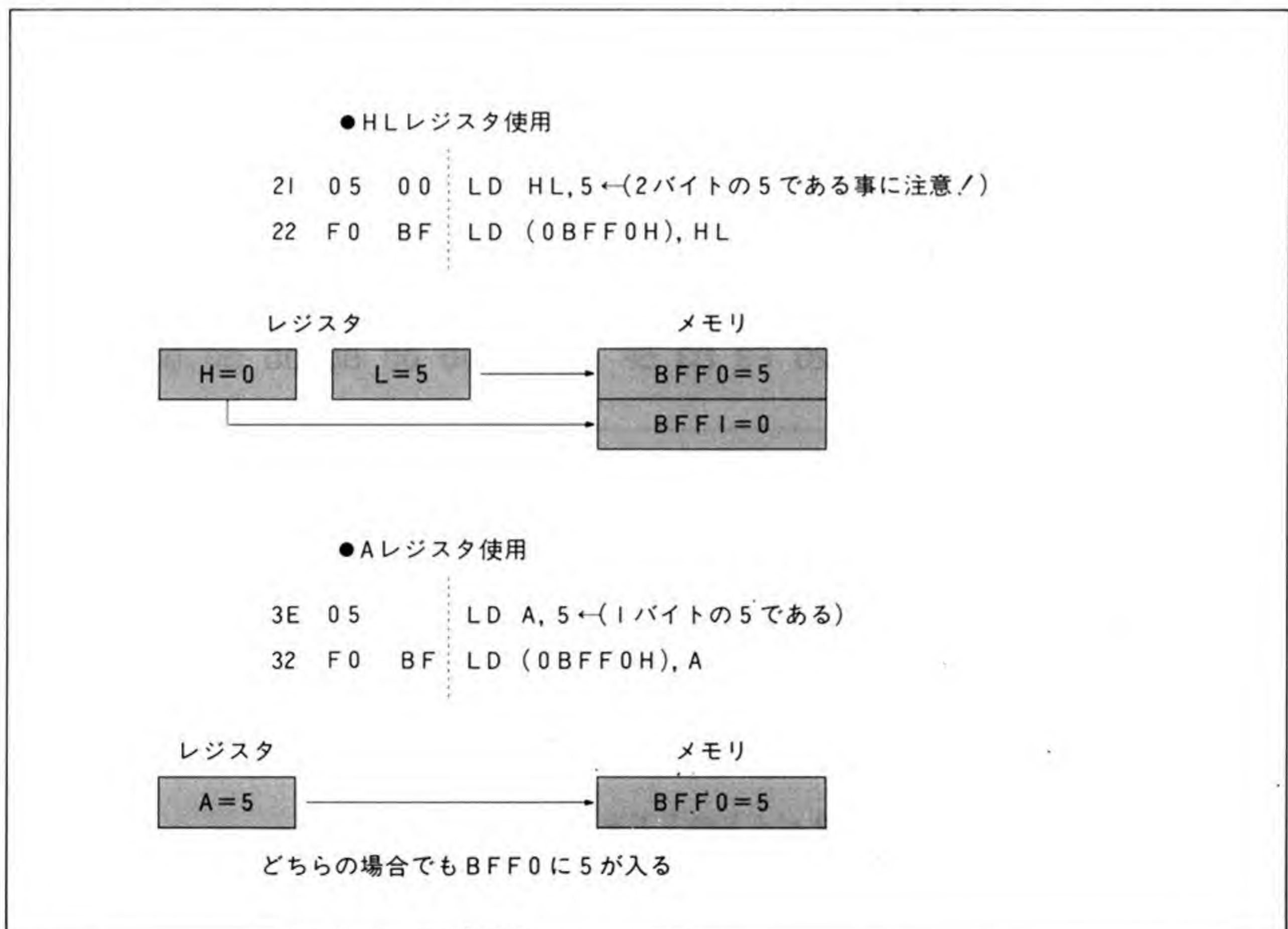


となり、同じ事でも4バイト使ってしまうわけです。

表中に(nn)とありますが、これは8ビットロード命令にも出て来ました。



第5-2図 2バイトのレジスタとメモリのやりとり



第5-3図 2バイトのロードと1バイトのロード



LD HL, (0BFF0H)

と書くと、BFF0番地の内容をHLレジスタに入れろと言う事になります。

これは変ですね。左側は2バイトで、右側が1バイト、これでは代入できません。

2バイトのロード命令では、メモリとのやりとりは第5-2図の様になります。つまり、HレジスタにBFF1番地の内容が、LレジスタにBFF0番地の内容が入るのです。BC、DEのレジスタペアでも同じ事が言えます。

2バイトの数が上下逆にコーディングされるというのは、単純に考えれば、番地の少ない方が下のバイトになると言うわけですが、ロード命令で考えると、第5-3図の様に、1バイトの数を2バイト命令でも、1バイト命令でも同じくロードできる事を意味します。

さて、1バイトのロードでは、

LD B, H

のように、それぞれのレジスタ間でロードができました。しかし、

LD BC, HL

と言う様な2バイトのロード命令はありません。

LD B, H

LD C, L

として、8ビットのロードを二回行う方法もありますが、16ビットのロード命令には、PUSH・POPと言う重要な命令があり、これを使う方法があります。

PUSH HL

POP BC

この2行の作業により、BCレジスタの中に、HLレジスタの中味が入ります。

PUSH・POP命令はBASICにはない考え方で、マシン語を使う上では、ぜひ知っておかねばなりません。それにはまず、スタックポインタ(SP)について知る必要があります。

#### ●スタックポインタとは

BASICで“GOSUB”文と言うのがありますね。マシン語では“CALL”を使いますが、どちらもサブルーチンをコールします。プログラムはサブルーチンの作業を終了すると、サブルーチンを呼び出した所に戻って、次の作業へ移ります。

従って、どこからサブルーチンへ飛んだのかをメモしておく必要があるわけです。そのメモを書いておく所が“スタック”で、Z80はスタックを一般のメモリ上に持ちます。その理由は、サブルーチン内で別のサブルーチンを呼び出すような作業が続けば、広いスタックエリアを必要とするからです。

スタックのある場所を示すのが“スタックポインタ”で、64K中どの番地にあるかを記録するために、2バイトのレジスタとなっています。

CPUはプログラムカウンタ(PC)と言うレジスタの示す場所の作業を行っているのですが、サブルーチンを呼び出す時は、このプログラムカウンタを一時退避させておいて、サブルーチンが終ると、退避させたプログラムカウンタを元に戻します。

同じ様にして、プログラムカウンタ以外の他の2バイトのレジスタも、スタックエリアに退避させる事ができます。しかも、どこで何のレジスタを退避させ、またスタックから取り出すかは、

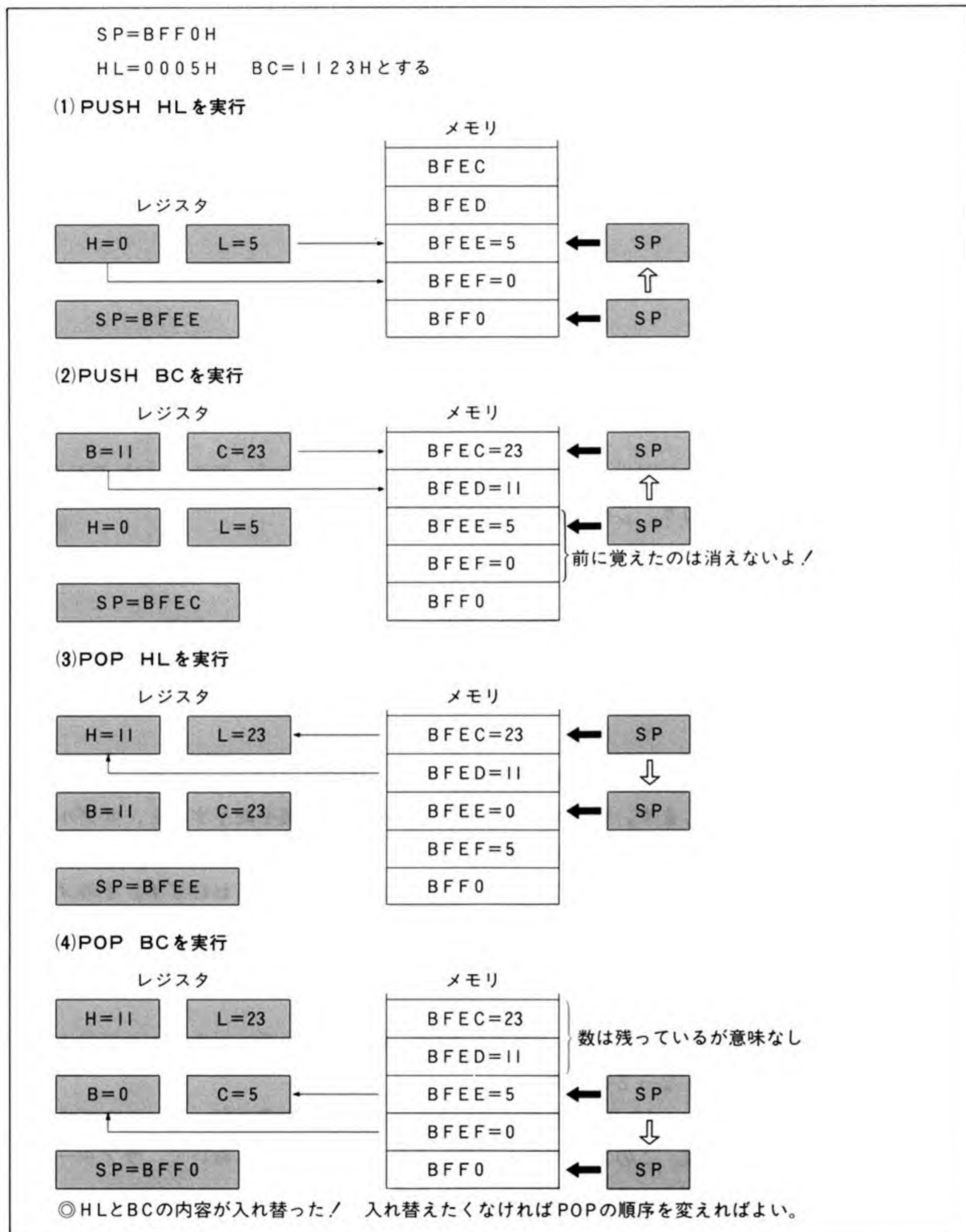


プログラムでできます。

その命令がPUSH・POPなのです。

たとえば、

PUSH BC



第5-4図 SP(=BFF0H)の場合のPUSH・POP



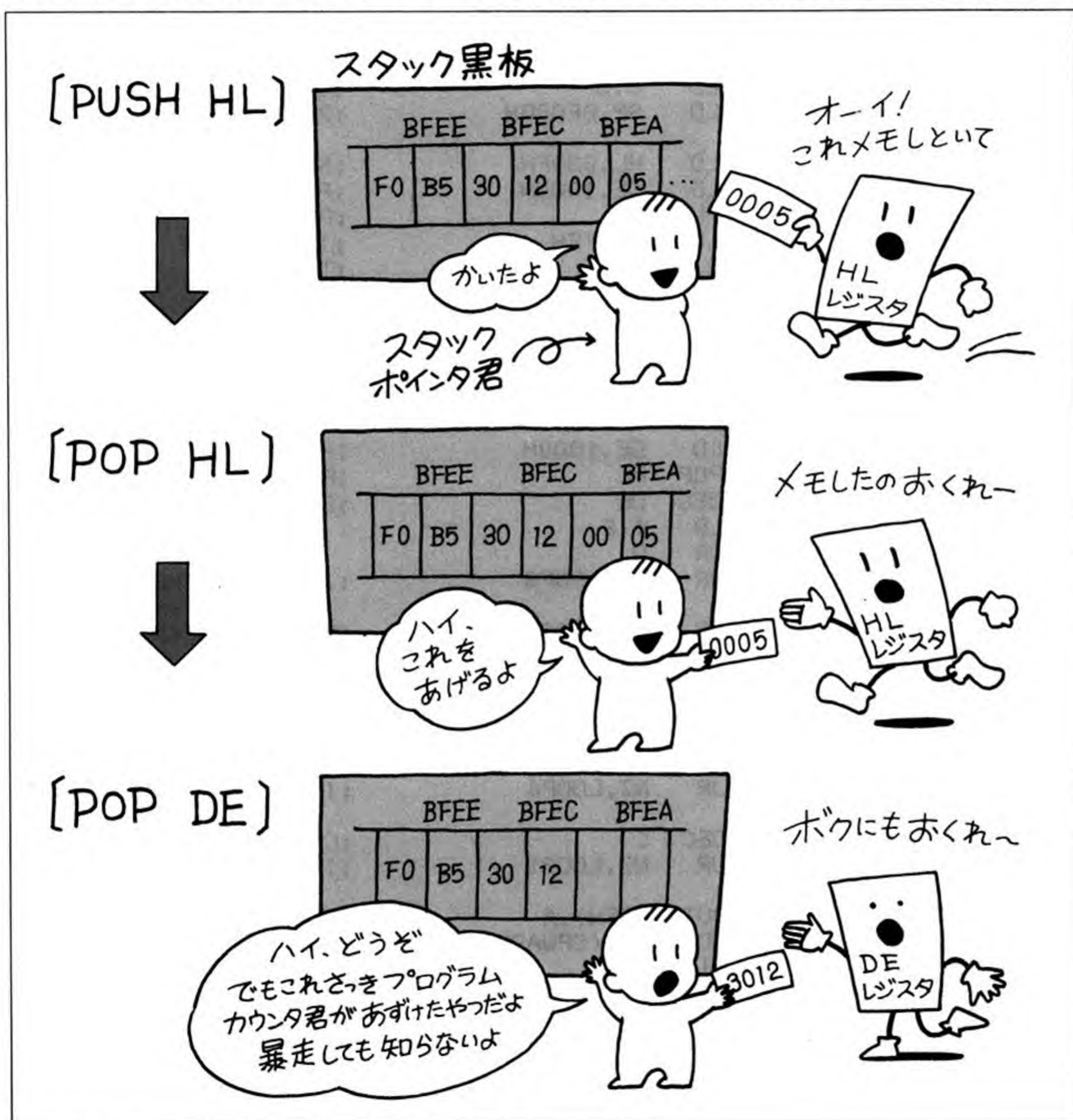
と命令すると、BCレジスタの内容がスタックエリアのメモリの中に入ります。

ですから、PUSH・POP命令はレジスタとメモリとの間のロード命令と考えてよいのです。

ただし、LD (0BFF0H), HL 命令が、BFF0番地にL, BFF1番地にHレジスタの値が入ったのに対して、PUSH HL では、スタックポインタの示す番地の一つ前にH, 二つ前にLレジスタの値が入ります。そして、自動的にスタックポインタの値は二つ減り、次にPUSH命令が行われても、前に退避した値は保存されます。

第5-4図はスタックポインタ(SP)の値が、BFF0の時のPUSH・POPの様子を示しています。

このように、スタックに値が送り込まれた時、どのレジスタから来た値かを記録しているわけではなく、入った順に積まれて行き、POP命令があると上から順に出して行きます(第5-4図





および第5-5図)。

今、あなたのパソコンのスタックエリアがどこにあるかわかりますか？

それを見るには、モニタの“X”コマンドを使います。

#### リスト5-3a PUSH POPテストプログラム

```

;
; PUSH POP テスト
;
;
;          ORG  0BF00H
;
BF00 F3      PPT:  DI          ;ワリコミ キンシ
BF01 ED73CBF LD      (SPWARK),SP ;スタック ポインタ ホソン
BF05 D35E    OUT     (5EH),A    ;SELECT GREEN G RAM


BF07 0E03    LD      C,3        ;3 カイ
BF09 3100F0  LOOP1: LD      SP,0F000H ;スタック エリア MAX SET
;
BF0C 210F03    LD      HL,030FH  ;SET BIT PATTERN
BF0F 110010    LD      DE,1000H  ;PUSH カイスク
BF12 E5      LOOP2: PUSH HL      ;PUSH !!!
BF13 06FF      LD      B,0FFH    ;TIMMER SET
BF15 10FE      WAIT: DJNZ WAIT   ;TIMMER
BF17 1B        DEC      DE        ;DE=DE-1
BF18 7B        LD      A,E
BF19 B2        OR      D
BF1A 20F6      JR      NZ,LOOP2   ;IF DE<>0 THEN LOOP2
;
;イカ ハ LD SP,0F000H テモヨイ
BF1C 110010    LD      DE,1000H  ;POP カイスク
BF1F E1      LOOP3: POP      HL   ;POP !!!
BF20 1B        DEC      DE        ;DE=DE-1
BF21 7B        LD      A,E
BF22 B2        OR      D
BF23 20FA      JR      NZ,LOOP3   ;IF DE<>0 THEN LOOP3
;
BF25 210000    LD      HL,0000H  ;SET 0 (CLEAR)
BF28 110010    LD      DE,1000H  ;PUSH カイスク
BF2B E5      LOOP4: PUSH HL      ;PUSH !!!
BF2C 1B        DEC      DE        ;DE=DE-1
BF2D 7B        LD      A,E
BF2E B2        OR      D
BF2F 20FA      JR      NZ,LOOP4   ;IF DE<>0 THEN LOOP4
;
BF31 0D        DEC      C          ;C=C-1
BF32 20D5      JR      NZ,LOOP1    ;IF C<>0 THEN LOOP1
;
BF34 D35F      OUT     (5FH),A    ;RAM
BF36 ED7B3CBF LD      SP,(SPWARK) ;スタック モトス
BF3A FB        EI              ;ワリコミ カイショ
;
BF3B C9        RET              ;ブレーク ポイント ノ カワリ
;
BF3C 0000      SPWARK:DW 0        ;スタック ホソン ハショ
;
BF3E          END

```




## リスト5-3b 実行プログラム


```
BF00 F3 ED 73 3C BF D3 5E 0E 03 31 00 F0 21 0F 03 11
BF10 00 10 E5 06 FF 10 FE 1B 7B B2 20 F6 11 00 10 E1
BF20 1B 7B B2 20 FA 21 00 00 11 00 10 E5 1B 7B B2 20
BF30 FA 0D 20 D5 D3 5F ED 7B 3C BF FB C9 00 00 00 00
```

h] x 

を実行するだけで、各レジスタの値を見る事ができます。SPの値も見つかりましたね。

h] ^ B 

でBASICへ戻し、

CLEAR , &H×××× 

として、適当な数字でクリア文を行うと、SPの値が変化するのが見られます。

では、SPをGRAM上に置いてスタックが積まれる様子を目で見る事にしましょう。

リスト5-3を入力し、

h] GBF00 

を行うと、画面に縞模様が現れて消えます。タイマーを取るともっと速くなります。画面クリアなどではPUSH命令を使うのが最も速いようです。

いかがですか、PUSH・POP命令について、おわかりいただけましたでしょうか？よくわからなくても、少ないレジスタを活用するために、バンバン使って下さい。

ただし、慣れないうちは次の点に気をつけて使うと良いと思います。

## ●PUSH・POP使用上の注意

(i) スタックポインタの位置を、C000番地より前にして、バンク切り替えの影響を受けないようにする。

ただし、スタックポインタの前、数十バイトは空いている事。また、BASICに戻りたい時などは、リスト5-3でやったように、SPを元に戻してからマシン語の作業を終える。

(ii) 原則として、PUSHしたらPOPするように、常にPUSH・POPを対応させて、PUSHしっぱなしで他の作業へジャンプさせたりしない（多くの場合暴走します。できればPUSHしたレジスタにPOPするようにした方がミスが少ない）。

16ビットのロード命令にはこの他レジスタの変換命令というのがありますが、内容が少なく、私は、

EX DE, HL (DEレジスタとHLレジスタの中味をそっくり交換する)

というものの位しか使っていません。(付録C表参照)

## 5-6

## “旗(フラグ)”の事

BASICでは“IF”という条件文があります。条件文はプログラムの流れを決める大切なものです。ゲームでも、敵にタマが当たったかどうか判定するのに、条件文なしでは考えられません。



マシン語では条件文は“フラグ”によって働きます。

フラグ(F)レジスタの事は前に書きましたが、この8ビットのうち、6ビットが一つ一つ独立したフラグとなっています。つまり、Fレジスタは、8ビット全体の数値としては無意味なのです。

条件文はFレジスタ内の決められたフラグの上げ降ろし(ビットのON, OFF)を見て、プログラムの行方をスイッチします。

六つものフラグを扱うのは大変です。しかし、実際に、ゲームプログラムで使うのは、このうち二つで足りるので、あとの説明は省略します。

### ●ゼロ(Z)フラグ

CPUがある計算をした時、結果が0になるとゼロフラグがセットされます。

### ●キャリー(CY)フラグ

CPUがある計算をした時、結果がマイナスになったり、けたあふれ(オーバーフロー)した時、キャリーフラグがセットされます。たとえば足し算をして、1けたと1けたを足すと1けたかもしくは2けたに(けた上り)なりますが、それはキャリーフラグを見ればわかります。キャリーは記号で“CY”と表しますが、これは“C”レジスタと混同しないようにするためです。ただし、ニーモニックの条件命令では“C”だけで表します。

さて、前に書いたように、フラグはFレジスタ上のビットです。したがってフラグがセットされると言う事は、フラグの値が“1”になると言う事です(ビットの値は0か1)。

では、このフラグはどのように利用されるのでしょうか。プログラムでいちいちFレジスタの内容を見に行く必要があるのでしょうか。その必要はありません。計算をしたり、ジャンプさせたりする命令の中で、CPUが見に行くものなのです。

フラグはCPUが何かするたびに、パタパタと変化(0と1の間で)しますから、プログラマが気をつけなければいけないのは、どの命令でフラグが変化し、どの命令では変化しないか、です。

リスト5-2や5-3に

DEC DE (DEを一つへらす) .....①

LD A, E (AにEを代入) .....②

OR D (AとDの論理和をとる) .....③

JR NZ, ~ (0でなければ ~ へ行く).....④

と言う部分がありますね。②をのぞいて、まだ説明していない命令ばかりですが、簡単にカッコをつけて説明しておきました。

このプログラムの目的は、DEレジスタの値を一つ減らして、0にならないければ、~にジャンプさせる事です。どうして、

DEC DE

JR NZ, ~

としないのでしょうか？

その理由は、16ビットのDEC命令では、どんな場合でも、フラグの変化が起きないのです！(8ビットのDECはZフラグ変化)

このように、常にフラグが変化するとは限らないので、付録の表を見てフラグ変化を確認しながらプログラムします(付録A表中のフラグ変化参照)。



## 5-7 計算させる方法

コンピュータは計算機です。したがって、CPUに計算させる命令も多数あります。

しかし、あらかじめ申しますと、“大した計算”はできないのです。BASICでは、かけ算わり算を始めとして、指数計算や三角関数も計算できますが、それは、プログラムで計算できるように作ってあるからで、CPUがいきなり三角関数を計算するわけではないのです。

CPUのする演算は、足したり引いたり、2倍したり、2で割ったりと言った単純なものばかりです。しかし、ゲームを作るのに“大した計算”は必要ありません。かえって単純なだけ親近感を覚えました、私の場合。

では、まず8ビットの計算から見て行きましょう。

付録C表に8ビットの算術演算をまとめたものがあります。このうち“INC”“DEC”命令をのぞいて、全部Aレジスタに対して行われるものなのです。

たとえば、先頭の“ADD”命令は、Aレジスタと表の横に並べたレジスタとの間で行われ、結果はAレジスタに入ります。

そこで、ニーモニックの表現は次のようになります。

オブジェクト	ニーモニック
8 7	ADD A, A
8 0	ADD A, B
8 1	ADD A, C
8 2	ADD A, D
⋮	⋮

しかし、“AND”命令や、“CP”命令などでは、始めの“A”はつけません。

AND A, B → AND B

SUB A, L → SUB L

CP A, D → CP D

と書きます。その理由は、付録C表に16ビットの算術命令がありますが、そちらと間違えないため、アセンブラを使う場合には気をつけなければなりません。16ビットに同じ命令のあるものは“A”をつけ、ないものはつけません。まあ、ハンドアセンブラのうちは書き方にこだわるよりも、まずプログラムができる方が先だと、と思いますが……。

各命令の内容を第5-3表にまとめておきましたので見て下さい。

論理演算について例を上げてみます。

Aレジスタが55H、Bレジスタが93Hのとき、

AND B をすると、ビットの様子は、

A : 55H → 0 1 0 1 0 1 0 1

B : 93H → 1 0 0 1 0 0 1 1

ANDは11H ← 0 0 0 1 0 0 0 1

になります。この11HがAレジスタに入ります。

同じ数字で、OR Bをした時は、



第5-3表 8ビット算術演算命令の働き

ニーモニック	働 き
ADD A, S	$A \leftarrow A + S$ 単純な足し算
ADC A, S	$A \leftarrow A + S + CY$ 上の計算にさらにキャリーフラグも足す。 上の位の足し算で使うと下のケタ上りが自動的に足せる。
SUB S	$A \leftarrow A - S$ 単純な引き算
SBC A, S	$A \leftarrow A - S - CY$ 上の計算にさらにキャリーフラグも引く。 上の位の引き算で使うと下のケタのケタ上がりが自動的に行える。
AND S	$A \leftarrow A \wedge S$ 論理積 両方ともにビットがONになっている所だけがONになる。
OR S	$A \leftarrow A \vee S$ 論理和 少なくとも片方のレジスタのビットがONになっていればそのビットはON。
XOR S	$A \leftarrow A \otimes S$ 排他的論理和 片方のレジスタのビットがONになっていれば、ONだが、両方のビットがONになっている所はOFFになる。
CP S	$A - S$ 仮想引き算または比較。 フラグは変化するが、結果はAに入らない。
INC m	レジスタの内容やメモリの内容が一つふえる。
DEC m	レジスタの内容やメモリの内容が一つへる。

表中、Sはレジスタ、具体的な数字、メモリの内容を表す。mはレジスタ、メモリの内容を表す。

A : 55H → 0 1 0 1 0 1 0 1  
B : 93H → 1 0 0 1 0 0 1 1  
ORはD7H← 1 1 0 1 0 1 1 1

となります。  
また、同じ数字で、XORはどうなるでしょうか、

A : 55H → 0 1 0 1 0 1 0 1  
B : 93H → 1 0 0 1 0 0 1 1  
XORはC6H← 1 1 0 0 0 1 1 0

となります。  
算術演算は、Aレジスタ同志でも行えます。  
たとえば、ADD A, Aとすると、AにAを足すわけですから、2倍のAがAレジスタに入るわけです。  
また、AND Aや、OR Aでは、Aレジスタの値は全く変化しません。一見無意味に見えますが、キャリーフラグを0にしたい時などに使います。  
XOR Aでは、同じもの同志でXORをするのですから、Aの中味は0になります。LD A, 0を行うよりも少ない命令でAを0にできるのです。  
では、計算の結果を目で見てみましょう。  
リスト5-4は、リスト5-2を少し書き替えたものです。



## リスト5-4a 8ビット演算命令テストプログラム

```

;
; 8 BIT インサ"ン メイレイ テスト
;
;
;          ORG  0BF00H
;
BF00 F3      CALC:  DI                      ;ワリコミ キンシ
BF01 ED7337BF LD    (SPWARK),SP              ;スタック ホ"インタ ホソ"ン
BF05 31F0BF   LD    SP,0BFF0H              ;スタック イト"ウ
BF08 D35E     OUT   (5EH),A                ;GREEN GRAM
;
BF0A 2100C0   LD    HL,0C000H              ;G RAM ADDRESS TOP
BF0D 110040   LD    DE,4000H              ;LD スル ナカ"サ
;
BF10 E5       PUSH HL
BF11 D5       PUSH DE                      ;DE,HL ホソ"ン
;
BF12 3E88     LOOP1: LD  A,88H              ;SET BIT PATTERN DATA
BF14 77       LD    (HL),A                ;SET DATA TO MEMORY
;
BF15 06FF     LD    B,0FFH
BF17 10FE     WAIT1: DJNZ WAIT1            ;WAIT TIMMER
BF19 23       INC   HL                    ;HL=HL+1
BF1A 1B       DEC   DE                    ;DE=DE-1
BF1B 7B       LD    A,E
BF1C B2       OR    D
BF1D 20F3     JR    NZ,LOOP1              ;IF DE<>0 THEN LOOP
BF1F D1       POP   DE                    ;DE,HL モト"ス
BF20 E1       POP   HL
;
BF21 7E       LOOP2: LD  A,(HL)            ;GET BIT PATTERN DATA
BF22 F60F     OR    0FH                  ;TEST [OR 0FH]
BF24 77       LD    (HL),A                ;SET DATA TO MEMORY
;
BF25 06FF     LD    B,0FFH
BF27 10FE     WAIT2: DJNZ WAIT2            ;WAIT TIMMER
BF29 23       INC   HL                    ;HL=HL+1
BF2A 1B       DEC   DE                    ;DE=DE-1
BF2B 7B       LD    A,E
BF2C B2       OR    D
BF2D 20F2     JR    NZ,LOOP2              ;IF DE<>0 THEN LOOP
;
BF2F D35F     OUT   (5FH),A              ;RAM
BF31 ED7B37BF LD    SP,(SPWARK)          ;スタック モト"ス
BF35 FB       EI                          ;ワリコミ カイシ"ョ
;
BF36 C9       RET                        ;フ"レーク ホ"イント ノ カワリ
;
BF37 0000     SPWARK:DW  0                ;スタック ホソ"ン ハ"ショ
;
;
BF39          END

```



## リスト5-4b 実行プログラム

```

BF00 F3 ED 73 37 BF 31 F0 BF D3 5E 21 00 C0 11 00 40
BF10 E5 D5 3E 88 77 06 FF 10 FE 23 1B 7B B2 20 F3 D1
BF20 E1 7E F6 0F 77 06 FF 10 FE 23 1B 7B B2 20 F2 D3
BF30 5F ED 7B 37 BF FB C9 00 00 00 00 00 00 00 00

```

h) G B F 0 0 

で実行すると、画面一杯に、B F 13 番地で指定した、Aレジスタのビットパターンを表示します。

さらに、B F 22 番地の演算命令に従い、B F 23 番地の数と、画面のビットパターンを演算したものを表示していきます。この場合はO R 命令ですが、命令の種類や、演算する数字をいろいろと変えて、目で確かめて下さい。

次に16ビットの算術演算命令を見てみましょう。8ビットと同様に、“I N C” “D E C” 命令以外はH L レジスタとの演算で、演算の結果はH L レジスタに入ります。ただし、“A D D” 命令ではH L レジスタの替りに、I X レジスタとI Y レジスタが使えます。

各命令の働きを第5-4表に簡単にまとめてあります。注意しなければならないのは、H L, I X, I Yの間では演算命令がない事です。

第5-4表 16ビット算術演算命令の働き

ニーモニック	働 き
<b>A D D H L, S S</b>	$HL \leftarrow HL + SS$ 2 バイトの単純な足し算
<b>A D C H L, S S</b>	$HL \leftarrow HL + SS + CY$ 足し算にさらにキャリーフラグ(0か1)を足す
<b>S B C H L, S S</b>	$HL \leftarrow HL - SS - CY$ 引き算にさらにキャリーフラグを引く。 ◎ Z 80には単純な2バイトの引き算がないので単純に2バイトの引き算を行いたい時も、この命令を使います。この命令を行う直前に[AND A]などを行って、CY (キャリーフラグ)を0にしてやれば問題ありません。
<b>I N C S S</b>	$SS \leftarrow SS + 1$ レジスタペアの内容が1ふえます。 たとえば、Hレジスタが1で、LレジスタがFFの時、[INC HL]を行うと自動的にくり上って、Hの値が2、Lの値が0になります。
<b>D E C S S</b>	$SS \leftarrow SS - 1$ レジスタペアの内容が1へります。
SSは、レジスタペア、BC, DE, HL, SP(スタックポインタ)を示します。命令によっては、インデックスレジスタIX, IYも使えます。	



## ●特別な演算

付録C表のAFレジスタ操作命令を見て下さい。

## (i) DAA命令

ゲームの得点および得点表示を行う場合、16進数で得点を計算をして、10進数に直して表示するというのはやっかいです。

10進数で計算できれば便利なのですが、普通は5と5を足すと10にはならず0Aと言う数字が入ってしまいます。16進法ですから当たり前ですね。ところが、5と5を足して10Hにする計算ができるのです！

LD A, 5 ——— ①

ADD A, 5 ——— ②

DAA ——— ③

の様にします。②は単純な足し算で、この段階ではまだAに0AHが入っています。

しかし、③の“DAA”と言う命令をつけると、Aの中味は10Hに変身します。この様に計算のすぐ後にDAA命令をつけるだけで10進数として計算ができます。

リスト5-5は、これを利用したもので、すぐにゲームに使えるプログラムです。

使い方は、BFF0番地からBFF2番地までを、あらかじめ00にしておき、HLレジスタの中に10進数2ケタの数を入れて、BF03をコールすると、BFF0番地からBFF2番地の値とHLレジスタの値を10進加算して戻ります。

プログラムの先頭に、テスト用に105をHLレジスタに入れてありますので、

## リスト5-5a 3バイト10進数得点計算プログラム

```

;
; 3 バイト 10進数得点計算プログラム
;
; WIDTH 80,25 ョウ ヒュウシ" X=0-5 Y=24
;
;
; ORG 0BFF0H
;
BF00 210501 LD HL,0105H ;TEST 105 テン
;
; POINT:
BF03 3AF0BF LD A,(0BFF0H) ;IN HL 2 バイト 10進数
BF06 85 ADD A,L ;1,2 ケタ
BF07 27 DAA ;A=A+L
BF08 32F0BF LD (0BFF0H),A ;10進 0セイ
;
BF0B 3AF1BF LD A,(0BFF1H) ;3,4 ケタ
BF0E 8C ADC A,H ;A=A+H+CY
BF0F 27 DAA ;10進 0セイ
BF10 32F1BF LD (0BFF1H),A
;
BF13 3AF2BF LD A,(0BFF2H) ;5,6 ケタ
BF16 2E00 LD L,0 ;L=0
BF18 8D ADC A,L ;A=A+CY (ケタアカリ ノミ)
BF19 27 DAA ;10進 0セイ
BF1A 32F2BF LD (0BFF2H),A

```



```

;
; スウシ" ヒョウシ" プログラム
;
BF1D 3AF0BF      LD  A,(0BFF0H)
BF20 E60F        AND  0FH                ;CUT 2 ケタ
BF22 C630        ADD  A,30H              ;アスキー コード" ニ スル
BF24 320DFF      LD  (0FF0DH),A          ;X=5,Y=24
BF27 3AF0BF      LD  A,(0BFF0H)
BF2A 0F          RRCA                    ;1 ケタ 2 ケタ イレカエ
BF2B 0F          RRCA
BF2C 0F          RRCA
BF2D 0F          RRCA
BF2E E60F        AND  0FH                ;CUT 1 ケタ
BF30 C630        ADD  A,30H              ;アスキー コード" ニ スル
BF32 320CFF      LD  (0FF0CH),A          ;X=4,Y=24
;
BF35 3AF1BF      LD  A,(0BFF1H)
BF38 E60F        AND  0FH                ;CUT 4 ケタ
BF3A C630        ADD  A,30H              ;アスキー コード" ニ スル
BF3C 320BFF      LD  (0FF0BH),A          ;X=3,Y=24
BF3F 3AF1BF      LD  A,(0BFF1H)
BF42 0F          RRCA                    ;3 ケタ 4 ケタ イレカエ
BF43 0F          RRCA
BF44 0F          RRCA
BF45 0F          RRCA
BF46 E60F        AND  0FH                ;CUT 3 ケタ
BF48 C630        ADD  A,30H              ;アスキー コード" ニ スル
BF4A 320AFF      LD  (0FF0AH),A          ;X=2,Y=24
;
BF4D 3AF2BF      LD  A,(0BFF2H)
BF50 E60F        AND  0FH                ;CUT 6 ケタ
BF52 C630        ADD  A,30H              ;アスキー コード" ニ スル
BF54 3209FF      LD  (0FF09H),A          ;X=1,Y=24
BF57 3AF2BF      LD  A,(0BFF2H)
BF5A 0F          RRCA                    ;5 ケタ 6 ケタ イレカエ
BF5B 0F          RRCA
BF5C 0F          RRCA
BF5D 0F          RRCA
BF5E E60F        AND  0FH                ;CUT 5 ケタ
BF60 C630        ADD  A,30H              ;アスキー コード" ニ スル
BF62 3208FF      LD  (0FF08H),A          ;X=0,Y=24
;
;
BF65 C9          RET                    ;ブレーク ポイント ノ カワリ
;
;
BF66            END

```

リスト5-5b 実行プログラム

```

BF00 21 05 01 3A F0 BF 85 27 32 F0 BF 3A F1 BF 8C 27
BF10 32 F1 BF 3A F2 BF 2E 00 8D 27 32 F2 BF 3A F0 BF
BF20 E6 0F C6 30 32 0D FF 3A F0 BF 0F 0F 0F 0F E6 0F
BF30 C6 30 32 0C FF 3A F1 BF E6 0F C6 30 32 0B FF 3A
BF40 F1 BF 0F 0F 0F 0F E6 0F C6 30 32 0A FF 3A F2 BF
BF50 E6 0F C6 30 32 09 FF 3A F2 BF 0F 0F 0F 0F E6 0F
BF60 C6 30 32 08 FF C9 00 00 00 00 00 00 00 00 00 00

```



## h] G B F 0 0 □

を実行すると、その度に B F F 0 番地から 3 バイトの値が、1 0 5 H ずつ増えて行きます。

実際にこれを表示するには、1 バイトの中味を上位と下位の二つに分けて表示します。つまり、3 バイトで 6 ケタの数が扱えるわけで、ゲームにはこれで十分でしょう。

ただし、6 ケタが一杯になると小さな数に戻ってしまいますから、999999 で止めるようにするか、一番上のケタがあまり上らないように、加える点数を加減します。

## (ii) C P L 命令

A レジスタのビットの ON・OFF を反転します。たとえば、A に 39 H が入っていて C P L を実行すると、

A : 3 9 → 

0	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---

 が、反転し、

A : C 6 ← 

1	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

 になります。

## (iii) N E G 命令

この命令は A レジスタの正負を反転します。今までは“負の数”と言うのは扱ってきませんでした。8 ビットの数は 0 から F F まで、正の数ですね。しかし、引き算の命令があると言う事は、結果がマイナスになる事も考えられるわけです。負の数は、次の「相対ジャンプ命令」にも出て来ますので、ぜひ知る必要のある事です。

普通、負の数を表す場合、“-”の符号をつけて表しています。では、マシン語ではどう表すのでしょうか？ ちょっと考えると“-”の符号をビットの中に入れる事はできそうにありません。ビットは 0 か 1 の数なのですから。

では、見方を変えて、引き算が“-”になる時、レジスタの中味はどうなっているのか見てみます。

L D A, 5

S U B 6

として、5 - 6 の計算を行ってみます。もし、5 - 5 ならば、当然 A レジスタの中味は 0 ですが、それよりさらに一つ多く引くと——“F F”になります。0 からいっぺんに全部のビットが ON の状態になり、この時、キャリーフラグも立ちます。

F F H を普通に考えれば“255”の事ですが、この場合、5 - 6 から出来た数字ですから、“-1”として考えて見ましょう。F F H = -1 と思い込むわけです。そう思いながら、次の計算をコンピュータにさせてみましょう。

L D A, 6

A D D A, 0 F F H

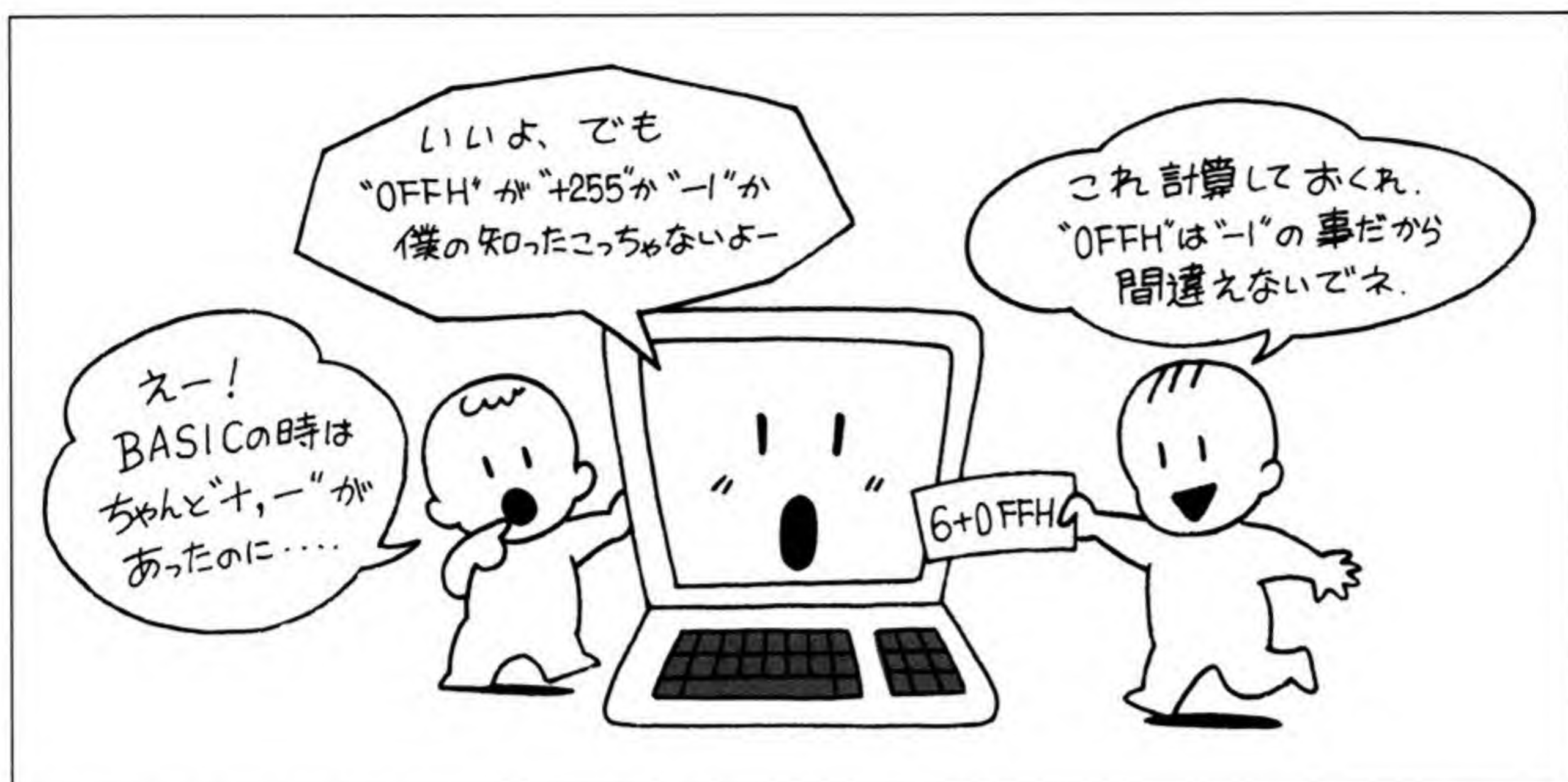
結果は、A レジスタに“5”が入り、この時もキャリーフラグが立ちます。

“6 - 1”をさせたわけですから当然です。では、“6 + 0 F F H”をさせた場合はどうなるのでしょうか？

1 バイトで計算させるなら、プログラムも計算も同じです！

いかがです？ 頭が混乱して来ましたか。しかし、混乱するのは人間だけでコンピュータは少しも悩みません。つまり、“0 F F H”を“255”と見るか、“-1”と見るかは人間がプログラムする際に考える事で、コンピュータにとっては問題ない、要は人間の見方なのです(第 5-6 図)。





第5-6図 マシン語の正と負

FFHが-1なら、FEHは-2、FDHは-3であると言えます。付録E表にこれをまとめたものがあります。

7FHより前は正の数、80Hより後は負の数として扱っていますね。つまり、一番上の第7ビットがONの時は負の数です。ですから、人間が数値を考える時、一番上のビットを正負を表すフラグと考え、“符号つき1バイト数”、“符号つき2バイト数”と呼ぶ事があります。

そして、Z80の命令にも正負の数を扱うための命令が少し用意されています。

NEG命令もその一つで、Aレジスタの正負を逆転させます。作業としては、CPLを行ってから1を足しています。なぜ1を足すか、例を見て下さい。

A: -3 → 

1	1	1	1	1	1	0	1
---	---	---	---	---	---	---	---

 (FDH)  
CPL A: 2 ← 

0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---

となって、ただ反転させただけでは3になりません。1を足せば反転させた数字になります。様々な場合について考えてみて下さい。

“符号つき数値”はむずかしいと思います。ゲームでは負の数を扱った方が便利な時もありますが、始めは正の数の範囲で考えるのが良いでしょう。

(iv) CCF命令      キャリーフラグを反転させます。

(v) SCF命令      キャリーフラグをON(1にセット)します。

## 5-8

### GOTO(ジャンプ)してみる

BASICでプログラムを組んだ事のある人ならば、GOTO文や、IF～THEN～文はご存知でしょう。プログラムの流れを決める非常に大切な命令で、これが出て来るといかにもプログラムらしくなりますね。

さて、Z80のジャンプ命令には大きく分けて二種類あります。



- (i)JP命令：具体的な2バイトの数で指定した番地にジャンプします(絶対ジャンプ)。
- (ii)JR命令：符号付きの1バイト数で指定した分だけの距離、その位置からジャンプします(相対ジャンプ)。

以上の2種類の中で、それぞれ各種の条件による命令に分かれます。これをまとめたものが、付録C表の、ジャンプ、コール命令です。

では、絶対ジャンプ(JP)命令について具体的に説明しましょう。

#### ●無条件絶対ジャンプ

GOTO文と同じで、文句なしに指定した所へプログラム(プログラムカウンタ)が移ります。

ニーモニック JP nn'

オブジェクト C 3 n' n

nn'は具体的な16進数で、ハンドアセンブルする時には、上下のバイトを逆にします。

#### ●ゼロフラグの内容によるジャンプ

BASICでは、

IF A=B THEN GOTO ~

と言う命令をよく使います。

マシン語では二つのレジスタの内容が同じかどうかを見る場合、(7)で登場した"CP"命令を使います。

CP B

を行うと、AレジスタとBレジスタの値が同じならばゼロフラグがONになります。

そこで、上のBASIC文は、マシン語だと、

CP B

JP Z, ~ (2バイトのアドレス)

となります。

"CP"の替りに"SUB"命令でもよいのですがAレジスタの値が変わってしまうので気をつけなければなりません。

AとBの値が違う時だけジャンプさせたい場合は、

IF A<>B THEN GOTO ~

マシン語では、

CP B

JP NZ, ~ (2バイトのアドレス)

となります。

以上のように、

JP Z, ~ (ジャンプ ゼロ)

JP NZ, ~ (ジャンプ ノンゼロ)

がゼロフラグによるジャンプ命令です。アセンブラを使ってアセンブルする時は、"カンマ"を忘れないで下さい。

さて、条件ジャンプは、もう一つ大切な働きがあります。それはループ作業での働きです。

FOR I=0 TO 5



(何かの作業)

NEXT I

と書けば、これもBASICでプログラムをした方は、よく使った事があると思います。

Iが0から5まで六回作業をくり返します。このプログラムを書き替えてみましょう。

I = 0

~ (何かの作業)

I = I + 1

IF I <> 6 THEN [前へジャンプ]

Iの値が5を越えて、6になるまで、FOR ~ NEXTと同じようにくり返し作業を行います。この様に、くり返し命令も条件文に直す事ができ、この形ならば、マシン語に置き替える事ができそうですね。

さらに、Iの値を5から始めて0で終わるようにするならば、マシン語プログラムはより作りやすくなります。

FOR I = 5 TO 0 STEP -1

~ (何かの作業)

NEXT I

これを書き替えて、

I = 5

~ (何かの作業)

I = I - 1

IF I <> 0 THEN [前へジャンプ]

変数IをAレジスタに置き替えて、マシン語にしてみましょう。

LD A, 5

~ (何かの作業)

DEC A

JP NZ, ~ [前へジャンプ]

マシン語とBASICがピッタリと対応していますね。1バイトの"DEC"命令では、減らしたレジスタの値が0になると、ゼロフラグがON(1)になります。ですから、"CP"などの命令を次にやらなくともよいのです。

ループ作業で注意しなければならないのは、ループ中の作業でループカウンタの値を変えてはいけないと言う事です。

BASICでも、Iをカウンタとして使い、作業中にIの値を変えてしまうとおかしくなりますね。

その心配をなくすには、カウンタを保存してしまえばよいのです。

LD A, 5

PUSH AF

~ (何かの作業)

POP AF



```
DEC A
JP NZ, ~
```

ただし、ジャンプ先に注意して、PUSHとPOPの中に割り込んだりしないように気をつけて下さい。

ループ作業の中に別のループを作ってもかまいませんし、あらかじめ設定していたループ回数よりも少ない回数で他の作業へ移ってもかまいません(第5-7図)。

### ●キャリーフラグの内容によるジャンプ

次の様な条件ジャンプをマシン語にするにはどうしたらいいでしょうか？

```
IF A ≥ 5 THEN GOTO ~
```

Aが5以上の時、~へジャンプするわけです。

そこで、Aと5の比較減算をやってみましょう。

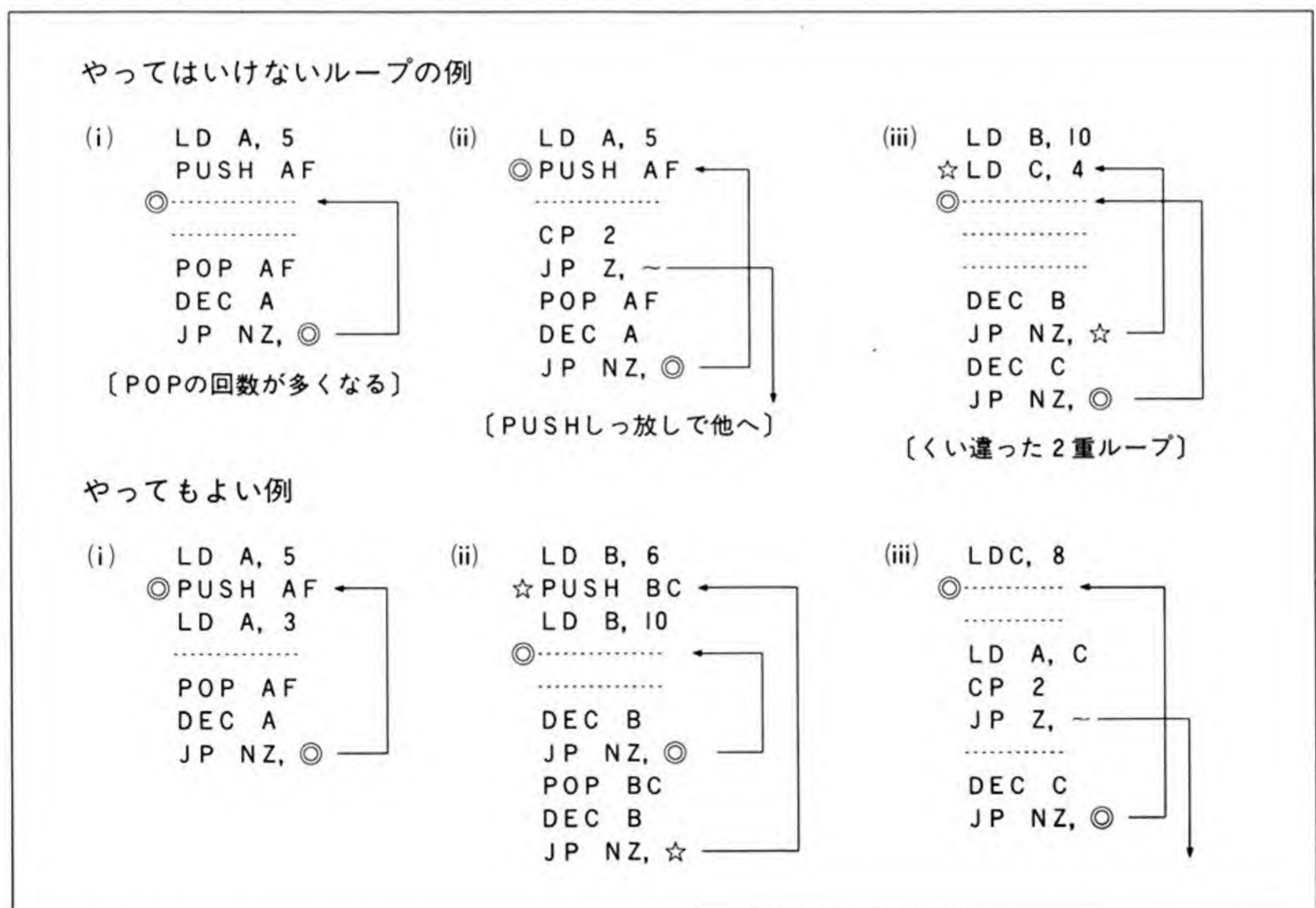
```
CP 5
```

ですね。もしAが6ならば、どのフラグもOFFですが、Aが5ならばゼロフラグが立ちます(ONになる)。しかし、この時キャリーフラグは立ちません。

もし、Aが4ならば、ゼロフラグは立たず、キャリーフラグが立ちます。

この様に、“CP 5”を行うと、Aが5から255の範囲ならば、キャリーフラグは立たず(0)、Aが0から4の範囲ではキャリーフラグが立ちます(1になる)。

つまり、「Aが5以上」と言う条件が成立するかどうか調べるためには、CP 5を行ってキャリーフラグがONになるかならないかで判断すればよいのです。



第5-7図 正しいループの使い方



Aが5以上の時はキャリーフラグは立ちません(0)。ですから、この場合は、キャリーが0、つまりノンキャリー(NC)の時に条件成立として、ジャンプさせます。すなわちマシン語では、

CP 5

JP NC, ~ (キャリーが0なら~へ)

となります。では、

IF A > 5 THEN GOTO ~

の場合はどうしたらよいでしょう。レジスタに入る数字は整数ですから、

IF A ≥ 6 THEN GOTO ~

と書き替えて考えればいいのです。

大小関係が逆の場合はどうでしょうか？

IF A < 5 THEN GOTO ~

Aが5より小さい時、条件成立ですから、条件の成立するような数字を考えて、たとえば、Aレジスタに3が入っている時、

CP 5

を行うと、3-5は負の数となるのでキャリーフラグが立ちます。すなわち、キャリーとなる場合に条件成立、ジャンプさせるので、マシン語は、

CP 5

JP C, ~ (キャリーが1なら~へ)

と言うプログラムになります。

条件付き絶対ジャンプ命令はこの他にもありますが、他のフラグについての説明を省きましたので、ここでは省略します。

次に相対ジャンプ命令について見て行きましょう。(i), (ii)で触れましたが、絶対ジャンプ(JP)命令が2バイトの数で、ジャンプ先の番地を指定するのに対し、相対ジャンプ(JR)命令は1バイトの数でジャンプ先を示します。

しかし、番地を示すには2バイトの数が必要です。どうやって1バイトで番地を示す事ができるのでしょうか。それを知るには、まずPC(プログラムカウンタ)について知らねばなりません。

CPUはメモリに書かれた命令や数値を順番に読み取り作業を行っていきますが、次にどの作業を行うのかを記録しているのが、プログラムカウンタです。

絶対ジャンプ命令により、作業が他のメモリ番地に書かれた所へ移るのは、プログラムカウン

プログラム							
番地	オブジェクト				ニーモニック		
BF 00	3E	15			LD	A, 15H	
BF 02	32	00	C0		LD	(0C0000H), A	
⋮	⋮	⋮	⋮		⋮		
メモリの中							
	BF 00	BF 01	BF 02	BF 03	BF 04	BF 05	⋯⋯⋯
	3E	15	32	00	C0		



## CPUの作業

CPU	プログラムカウンタ(PC)
3E 「Aに次の値を入れろ」を了解	BF00 → BF01
15 「15だぞ、と」Aに15を入れる	BF01 → BF02
32 「次のメモリにAの値を入れろ」を了解	BF02 → BF03
00 「C000だよ」C000番地にAの値を入れる	BF03 → BF04
C0	BF04 → BF05
[ここまでの作業が終わった所で プログラムカウンタの値はBF05になっている]	

第5-8図 CPUの作業とプログラムカウンタ

タの中に、ジャンプ命令による値が書き込まれるからです。

第5-8図はプログラムカウンタの変化を表したものです。一つの作業が行われると、プログラムカウンタは次の作業の番地を示します。

相対ジャンプ命令では、5-7の(iii)で説明したように、正負を考えた1バイト数(符号つき1バイト数)によって、現在のプログラムカウンタから数えて、-128から+127までの範囲でジャンプさせます。

言葉ではわかりにくいので、第5-9図を見て下さい。正の方向へのジャンプと、負の方向へのジャンプを例示してあります。

どちらも相対ジャンプ命令をCPUが受けとった後、プログラムカウンタの値は"BF07"です。ここを基準に考えなければいけません。

正方向のジャンプは、その値を決めるのが簡単で、"BF0A"へジャンプさせたい時は、プログラムカウンタの値を"3"ふやせばよいのですから、

18 03

となります。この時、LD HL, 0D000Hの命令は無視されて、先へ進む事になります。

逆方向への相対ジャンプはちょっと困りものです。"BF00"番地へ戻したいとすると、"BF07"から数えて七つ目になりますね。自分自身のJR命令も数えなくてはいけない所がクセモノです。

さて、"-7"は、付録E表により、"F9H"である事がわかりますから、その値を書き込みます。

18 F9

こうすると、プログラムは、BF00からBF06までの間を無限にくり返す事になります(無限ループ)。

今までに紹介して来ましたサンプルプログラムの中でも、条件付きの相対ジャンプによるループ作業が数多く見られますね。

ループに使われる場合は大抵逆方向のジャンプです。もし、アセンブラを使うのであれば、サンプルプログラムの様に番地を"ラベル"と呼ばれる名前で代用させておくと、オブジェクトを作



## ●正方向ジャンプ

番地	オブジェクト	ニーモニック
BF 00	3E 15	LD A, 15H
BF 02	32 00 C0	LD (0C000H), A
BF 05	18 <u>03</u>	JR ◎
BF 07	21 00 DD	LD HL, 0D000H
BF 0A	06 E9	◎LD B, 0E9H
⋮	⋮	⋮

(BF 0A)へジャンプ

(03)の計算法 BF 07 = 3

## ●逆方向ジャンプ

番地	オブジェクト	ニーモニック
BF 00	3E 15	☆LD A, 15H
BF 02	32 00 C0	LD (0C000H), A
BF 05	18 <u>F9</u>	JR ☆
BF 07	21 00 D0	LD HL, 0D000H
BF 0A	06 E9	LD B, 0E9H
⋮	⋮	⋮

(BF 00)へジャンプ

(F9)の計算法 BF 00 ~ BF 07 = -7     -7 ⇨ F9

第5-9図 相対ジャンプ命令

る時に自動的にジャンプ先までの値を計算してくれるのですが、ハンドアセンブルではそうは行きません。

第5-10図にハンドアセンブルのゴツのようなものを書いてみました。

実際には、プログラムを入力した後、モニタの逆アセンブル機能を使って、値が間違いない事を確認するのがよいと思います。

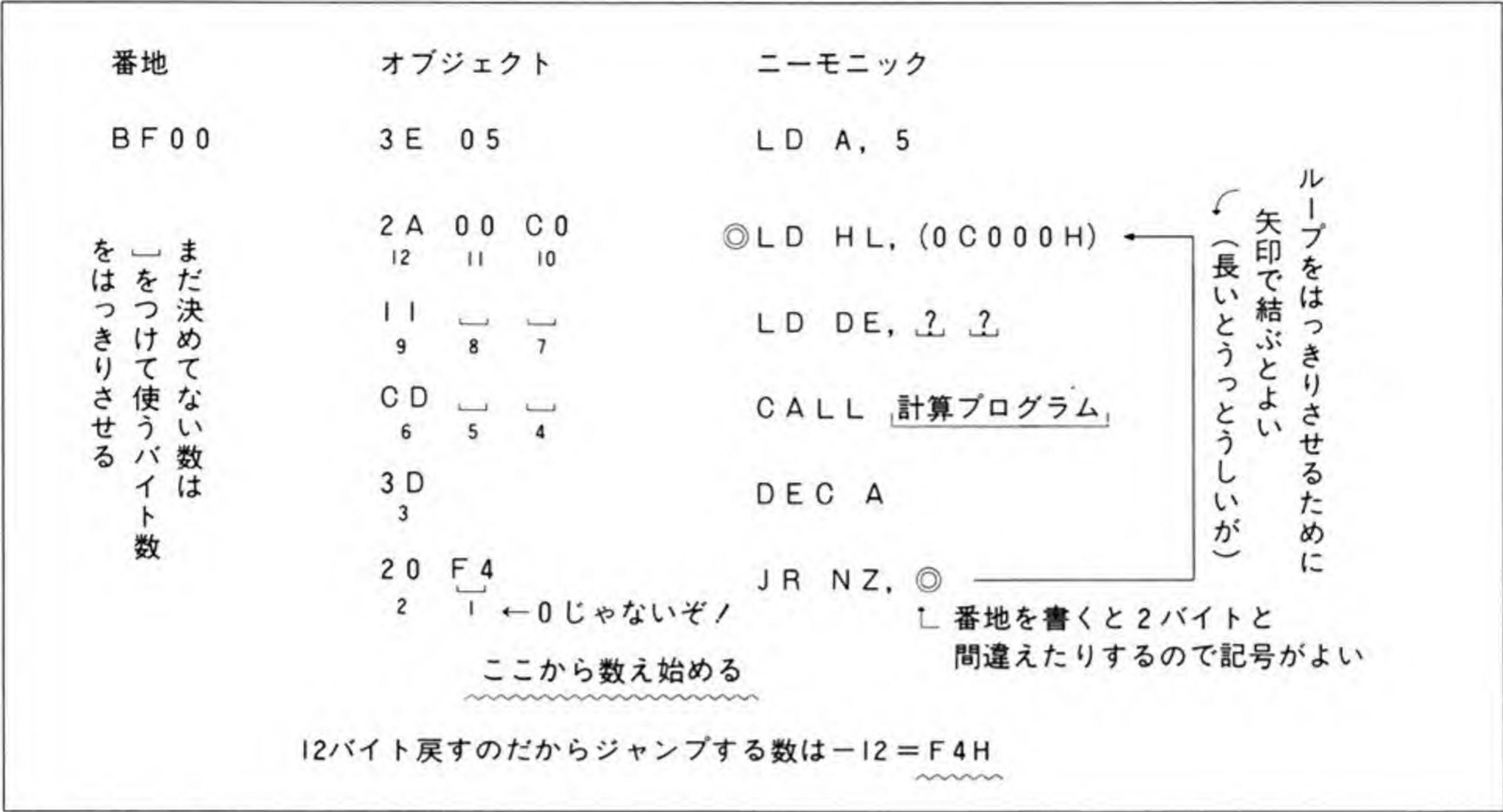
たとえば、第5-9図の逆方向ジャンプの例を入力し、モニタから、

h) LBF00

としてみて下さい。"L"は逆アセンブルするためのコマンドです。

```
BF00 3E 15      MVI A, 15
BF02 32 C000    STA C000
BF05 18 F9      JMPR BF00
⋮              ⋮
```





第5-10図 相対ジャンプ・ハンドアセンブルの例

と表示されるはずですが。ニーモニックが変な記号で並んでいますが、これはインテルのニーモニックです。

ザイログ社のZ80はニーモニックが異なり、命令も多く持っていますので、場合によっては“?”マークが出たりして、正確な逆アセンブルが行えないのですが、ここでの目的は~~~~部分, “JM PR BF00”と言う所を見る事なのです。

JM PRはJRと同じ意味で、次の番地がジャンプさせたい番地になっていればよいのです。この様に、相対ジャンプ命令は一見面倒くさそうな命令ですが、プログラムを作る位置を変えてもそのままの状態で見えますし、メモリも少なくて済みます。ただし、ジャンプできる距離が短いので、目で追える範囲のプログラム内で使うのが良いと思います。

条件付き相対ジャンプは、絶対ジャンプで説明したのと同じ物が使えます。

●特殊なジャンプ

(i)DJNZ命令

```
DEC B
JR NZ, ~
```

以上の二つの命令を同時にやってくれる命令です。Bレジスタのみの機能ですから、Bレジスタはループカウンタや、タイマによく使われます。

リスト5-2では、Bレジスタの値が0になるまでこの命令をくり返しやっていますね。他には何もしていないので、この場合は、単に時間待ちさせるだけに使っているわけです。便利ですから、ぜひ覚えましょう。

(ii)JP (HL) 命令

今までのジャンプ命令は全部、具体的な所へジャンプするものばかりでしたが、この命令はHLレジスタの中に入っている番地にジャンプするものです。



ジャンプ先がプログラムではわかりにくくなりやすいので、慣れないうちはあまり使わない方がよいと思います。

## 5-9 GOSUBは知っていますね？

BASICでサブルーチンを使った事がありますか？使った事のない方は、まずBASICでその使い方を学んで下さい。

プログラムを作る時、一番大切なのはサブルーチンの活用だと思います。

小さいプログラムに無理して使う事ありませんが、大きいプログラムになると、プログラムの流れがつかみにくくなり、バグも出やすくなります。

そこで、一つ一つの処理をサブルーチンとして完成させ、それが間違いなく働くならば、それらを組み合わせて大きなプログラムにすればすっきりとして、バグも起こりにくいものです。

第5-11図の様に、ゲームの場合でも、一つ一つの動きをサブルーチンにして組み立てます。したがってメインルーチンはサブルーチンを呼び出す（コールする）命令のかたまりとなります。

実際にはもう少し複雑になるケースが多いのですが、できるだけ一つのプログラムは短くし、サブルーチンの中にさらにサブルーチンを持たせるようにします。

たとえフローチャートをしっかり作ったにせよ、プログラムがゴチャゴチャで目で追えなくなると、作っていてもいやになってしまいます。

ところが、一つのまとまりをサブルーチンにして、別の場所に移してやると、見違えるようにすっきりとして来るのです。

前置きが長くなりましたが、BASICでサブルーチンを理解して下さいれば、マシン語でも使い方は同じです。

ただ、良い事に、マシン語のサブルーチンコールでは、ジャンプ命令の様に条件、つまりフラグによってコールするかしらないかを決める命令もあります。

同じ様にサブルーチンからリターンする時も、条件によってリターンするかしらないかできます。

これらの命令も、付録C表にまとめてあります。

書き方は、サブルーチンをコールするには、

CALL nn' → CD n' n

CALL Z, nn' → CC n' n

CALL NZ, nn' → C4 n' n

CALL C, nn' → DC n' n

CALL NC, nn' → D4 n' n

nn'は具体的な2バイトの番地です。

リターン命令は、

RET → C9 (無条件リターン)

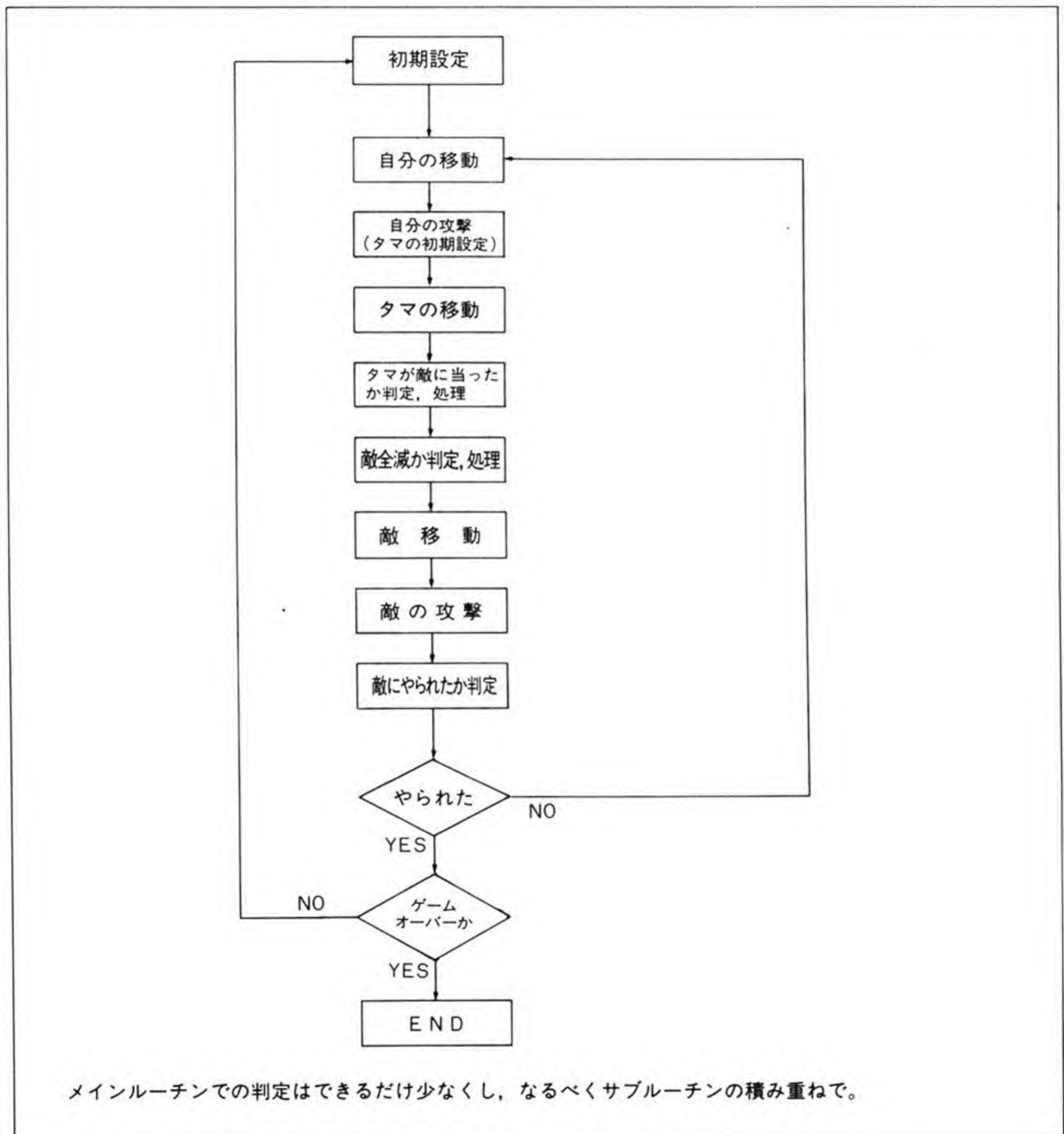


RET Z → C8 (ゼロフラグONの時RET)

以下略しますが、このようにリターン命令は1バイトで表せます。

CPUから見れば、CALL命令もJP命令も同じようなものです。ただCALL命令が行われると、スタックにプログラムカウンタの値を待避させておき、サブルーチンへジャンプさせ、リターン命令がくるとスタックから値を取り出して、プログラムカウンタに入れます。従って、サブルーチン中で、PUSH・POPの対応を間違えると元の所へ戻る事ができなくなるので注意しなければなりません。

逆にこの事を利用すると、“GOSUB 行番号” の様な事もできます。サブルーチンの終りに、POP HL (PCに戻るはずの値を横取り)



第5-11図 ゲームのメインプログラムフローチャート



LD HL, ~ (強制リターンさせたい番地)

PUSH HL (スタックに入れる)

RET (~へリターンする)

あまり使う事もないと思いますが、こんなテクニックもあります。

サブルーチンの良い所は、普通に使えば、どんな遠い所にサブルーチンがあっても、またそこに戻って来る事です。しかし、ジャンプでは、行きっぱなしですから、その後の事もずっと考えなければならないのです(第5-12図)。

くり返して書きますが、サブルーチンの重ね合わせ(構造化)によってプログラムを作るのが基本と言えます。

### PC(プログラムカウンタ)



### メインルーチン

どんな遠くのサブルーチンへ飛ぼうが、メインルーチンと言うオシャカ様の手から逃がれる事はできない。

第5-12図 サブルーチンコールとは



## 5-10

## ブロック転送は早いなのって

ゲームでは画面にキャラクタを書くために、多くのデータを移動する必要があります（テキストのキャラクタなら話しは別ですが）。

データを移動させる方法はいくつかあります。たとえば、HLで示される番地からDEで示される番地へデータを移し、HLの次の番地からDEの次の番地へデータを移し、これを五回くり返したいとします。キャラクタのPUTにはこの様な作業がよく出て来ます。

これをマシン語プログラムにしてみましょう。

LD B, 5	(カウンタセット)
☆ LD A, (HL) ←	(HL番地の内容をAに)
LD (DE), A	(Aの内容をDE番地へ)
INC HL	(HL←HL+1)
INC DE	(DE←DE+1)
DJNZ ☆	(5回くり返し)

カウンタのセットは別にして、これらの命令をなんと2バイトの1命令でやってしまう事ができます！“LDIR”と言う命令で、コードに直すと“ED B0”となります。

ただし、カウンタは、BCレジスタの値ですから、間違えて上の例のようにBレジスタに入れてしまうと、256倍もの数データを移動してしまいます。

このように、ブロック転送命令は、

HLレジスタ：元になるデータの先頭番地

DEレジスタ：複写したい所の先頭番地

BCレジスタ：カウンタ(データの量)

をセットして、その後にブロック転送命令を行うのです。

ブロック転送には、一回の命令で1バイトしか転送しないものや、HL、DEの値を減らして行くものなどがあります。これらは付録A表により機能を見て下さい。またマシンコードの一覧表は付録C表にあります。

ブロック転送がどのようなものか、目で見えるようにしたものが、一番初めに出来て来ましたリスト5-1です。リスト5-6にニーモニックで表したソースリストがあります。

このプログラムを実行すると、縞模様が画面に現れましたが、みなさんは何だと思われましたか？実はこの模様はPC-88のBASIC用ROMの内容や、ワークエリアの内容のビットパターンなのです。簡単にリストの説明をします。

DIは割り込み禁止です。通常グラフィック画面に切り替える時は、割り込みを禁止します。この理由および“割り込み”については、3章のBGMの所を読み直して下さい。

OUT命令は後で書きますので略します。

HLレジスタに8000Hを入れてスクロールスタート時の番地を指定しました。



## リスト5-6 ブロック転送ラストプログラム

```

;
;       フォックテンソク テスト プログラム
;
;       ORG 0BF00H
;
BF00 F3      BLOCK: DI                      ; フリコミ キンシ
;
BF01 AF      XOR  A
BF02 D351    OUT  (51H),A                  ; TEXT OFF
;
BF04 D35E    OUT  (5EH),A                  ; SELECT GREEN G RAM
;
BF06 210080  LD   HL,8000H                  ; テンソク モト ADDRESS
;
BF09 E5      LOOP: PUSH HL                  ; HL タイヒ
;
BF0A 1100C0  LD   DE,0C000H                  ; G RAM TOP ADDRESS
BF0D 010040  LD   BC,4000H                  ; テンソク ナカサ
BF10 EDB0    LDIR                          ; (DE)=(HL) HL=HL+1 DE=DE+1
;                                     BC=BC-1 BC=0 マテ
BF12 E1      POP  HL                        ; HL モトス
;
BF13 115000  LD   DE,50H
BF16 A7      AND  A                        ; CY=0
BF17 ED52    SBC  HL,DE                    ; HL=HL-DE-CY
BF19 18EE    JR   LOOP                    ; ムゲン ループ
;
BF1B FB      EI                            ; フリコミ キョカ
BF1C C9      RET                          ; RETURN (イミナシ)
;
;
;
BF1D         END                          ; リスト オフリ

```

DEレジスタにはグラフィック画面用RAMの先頭番地(C000)を入れます。ここが画面左上で、BCレジスタに4000Hを入れてブロック転送すると、8000HからBFFFH番地までのメモリの内容が画面一杯に現れます。

そこで、HLレジスタの値を元の8000Hに戻し、50Hを引いて、ふたたびC000から始まるグラフィックRAMに4000Hのデータをブロック転送します。

すると、データのパターンは先程のより1ライン(50Hバイト)下がって見られるので、これをくり返すと、ダウンスクロールが起こるのです。

おもしろいのは、グラフィック画面にすでにあるデータをブロック転送する所で、次第にスピードが速くなって行くような錯覚を覚えます。

ついでに、ブロックサーチ命令を紹介しておきます。あるメモリの中から、特定の数を見つける時に使われます。

たとえば、"CPIR"と言うニーモニックの命令は、HLレジスタにサーチしたいメモリの先頭を入れ、BCレジスタにサーチする長さを入れ、Aレジスタに探したい数を入れて、CPIR命令を行うと、AレジスタとHLで示されるメモリの内容が一致するか、またはBCレジスタの値が0になるまで、HLレジスタの値を一つずつ増やし、BCレジスタの値を一つずつ減らして行きます。

使わなくともよい命令なので(もちろん使えば速い時もある)くわしくは書きません。



## 5-11

### かけ算ができればいいなあ

10進法のかけ算で、ある数に10をかけるとどうなるでしょう。

12に10をかけると120、38に10をかけると380。つまり1ケタずれるわけです（アタリマエダネエ）。

では2進法で、ある数に2をかけるとどうなるでしょう。そう、やはり1ケタずれるのです。

$$38 = 26H = \boxed{00100110} \text{ —— ①}$$

$$38 \times 2 = 76 = 4CH = \boxed{01001100} \text{ —— ②}$$

①と②を比べれば明らかだと思います。全体が左へ一つ動いていますね。

では、割り算はどうでしょう。2進法で2で割る事は、右へ一つずらせばよいと言う事です。

このような作業を行うのが、“ローテート・シフト命令”です。

シフト命令にはキャリーフラグがからんで来ます。

#### ●RLC(RRC)命令

RLCは左へのシフト、RRCは右へのシフトの事です。方向が違うだけなので、RLCの方を例に見てみましょう。図で見るのがわかりやすいと思います。第5-13図で、第0ビットから第6ビットまでは、一つ上のケタ、つまり左へシフトするだけですが、第7ビットの中味は、第0ビットへとローテートし、同時に、キャリーフラグもこれと同じ状態にします。

#### ●RLA(RRA)命令

第5-13図下段は、キャリーもふくめた形のシフト命令です。

あらかじめ、キャリーを0にしてから、RLA命令を行うと数は2倍に、RRA命令を行うと2分の1になります。キャリーに関係なく、0を押し込めて2倍あるいは2分の1にするのがSLA, SRL命令です。これら他の命令の動作の仕方は、付録A表のオペレーション略図を見て下さい。

RLAの命令を使って2バイトの数を2倍にしてみましょう。

例としてHLレジスタに386Hを入れてみます。Lレジスタにはいる86Hは2倍するとケタ上りしますから、それを考えなければなりません。

AND A (キャリーフラグを0にする)

RLA L (下1バイトをシフト)

RLA H (ケタ上りがキャリーを通して自動的にHレジスタに入る)

これを図でみると、第5-14図のようになります。

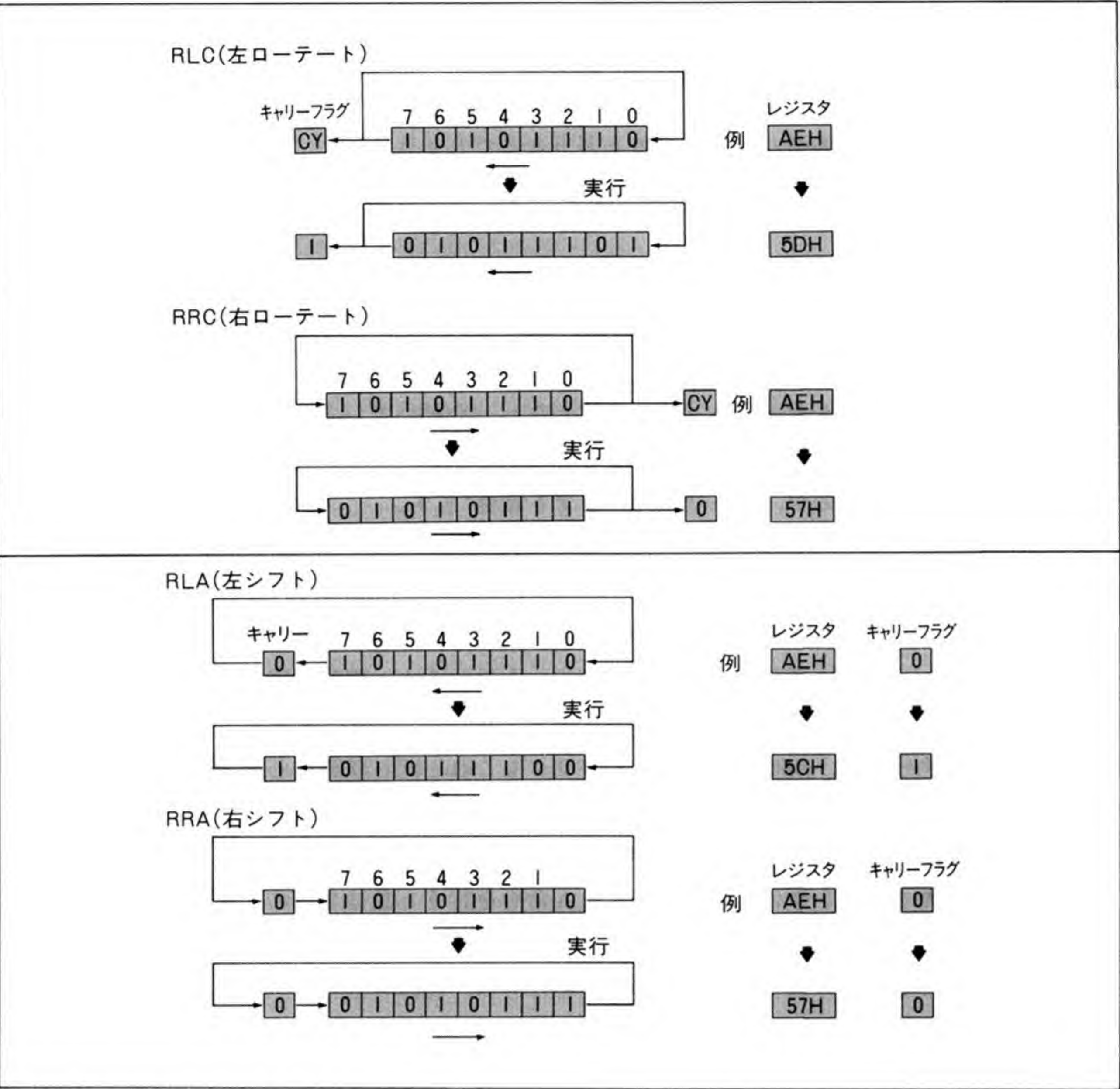
マシン語コードに直すには、付録C表を見て下さい。Aレジスタの四つのシフト命令は1バイトですみませんが、それ以外は2バイト以上の命令です。

シフト命令はただ2倍、2分の1するだけでなく、リスト5-5にあるように、RRC Aを四回行って上位と下位を入れ替えるような事もできます。

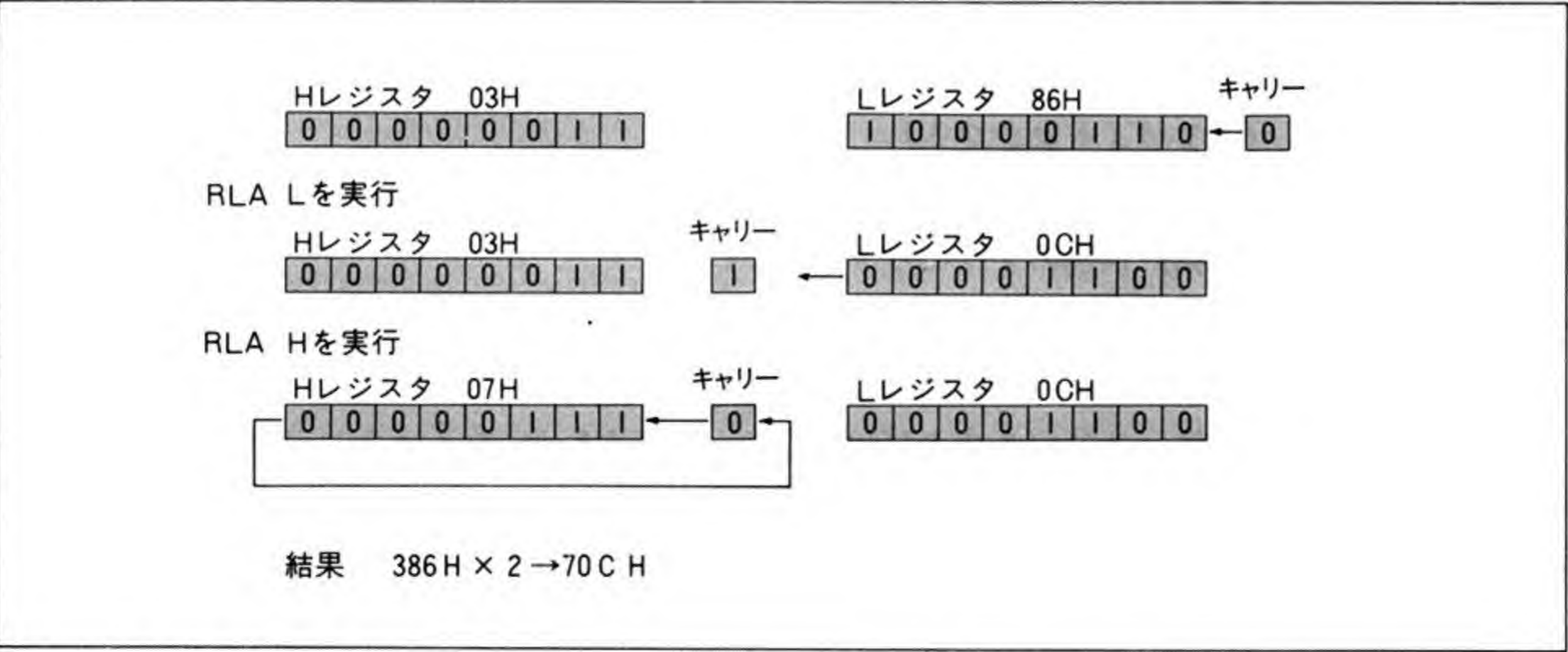
#### ●2倍以上のかけ算

3倍や5倍など、2倍以上のかけ算はどうすればいいのでしょうか。そんな計算やらないゲームを作ればいいのですが、なかなかそうもいきません。





第5-13図 ローテート、シフト命令



第5-14図 2バイト数の2倍



## リスト5-7a 2倍以上のかけ算プログラム

```

;
; L カケル H      IN H,L      OUT HL
;
;
;          ORG  0BF00H
;
BF00 213412      LD  HL,1234H      ;TEST DATA
;
BF03 F5          KAKERU:PUSH AF
BF04 C5          PUSH BC
;
BF05 AF          XOR  A            ;A=0
BF06 0608        LD   B,8          ;8 カイ シフト
BF08 CB1D        KAKER1:RR  L      ;L ミキ" シフト
BF0A 3001        JR   NC,KAKER2    ;ソノ ケタ カ" 0 ナラ シ"ャンプ°
BF0C 84          ADD  A,H          ;1 ナラ A ニ H ラ タス
BF0D 1F          KAKER2:RRA        ;A ミキ" シフト
BF0E 10F8        DJNZ KAKER1
BF10 CB1D        RR  L            ;L ミキ" シフト
BF12 67          LD   H,A          ;H ニ A ラ イレル
;
BF13 C1          POP  BC
BF14 F1          POP  AF
;
BF15 22F0BF      LD   (0BFF0H),HL  ;TEST ヲテI SET
;
BF18 C9          RET
;
;
BF19            END

```

## リスト5-7b 実行プログラム

```

BF00 21 02 12 F5 C5 AF 06 08 CB 1D 30 01 84 1F 10 F8
BF10 CB 1D 67 C1 F1 22 F0 BF C9 00 00 00 00 00 00 00

```

足し算をくり返しやる手もありますが、リスト5-7に例をあげておきます。LレジスタとHレジスタにかけ合わせたい数を入れて、

h) G BF03 


とすると、HLレジスタにかけ算の答えを入れて戻ります。このプログラムでは確認のためにBFF0番地に答えを入れています。

このプログラムは筆算のアルゴリズムをうまく使っています（マイコン82年7月号“野本さんの解答”より）。

なお、テスト用に、BF00から3バイトは、

LD HL, 1234H ⇒ 21 34 12

としてあります。つまり、BF01とBF02にかけあわせたい16進数を入れて、

h) G BF00 

とすると、答えがBFF0から2バイトに入ります。

## ● ランダムテクニック



このかけ算を利用して、ランダムナンバーを作り出す事を考えてみましょう。

BASICでランダム数を作る時は、

INT(A \* RND(1))

として、Aにある整数を入れると、0からA-1までの乱数が得られます。これは、RND(1)が、1以下の小数の乱数であるためです。

そこで、マシン語でも同じように考えると、0からFFHまでの乱数に、ある1バイトの数をかけ合わせます。答えは2バイトで得られますが、その上の部分だけをとると、0からある数-1

#### リスト5-8a 8ビットランダムメイカプログラム

```

;
; 8 BIT ランダム メイカ-      IN C      OUT 0 から C-1
;               ランダム データ- 1000H-1800H / ROM DATA
;
;               ORG  0BE00H
;
BE00 0E10      LD   C,10H                ;TEST 10H
;
BE02 F5        RAND: PUSH AF
BE03 D5        PUSH DE
BE04 E5        PUSH HL
;
BE05 2A34BE    LD   HL,(RAND0)           ;READ DATA ADDRESS
BE08 110018    LD   DE,1800H             ;DATA END
BE0B 7C        LD   A,H                  ;LIMIT CHECK
BE0C BA        CP   D
BE0D 3803      JR   C,RAND1
BE0F 210010    LD   HL,1000H             ;DATA TOP
BE12 23        RAND1: INC HL              ;DATA ADDRESS +1
BE13 2234BE    LD   (RAND0),HL
BE16 ED5F      LD   A,R                  ;READ リフレッシュ レジスタ
BE18 D5        PUSH DE
BE19 5F        LD   E,A
BE1A 1600      LD   D,0
BE1C 19        ADD  HL,DE                 ;DATA ADDRESS +R (0-7FH RND)
BE1D D1        POP  DE                   ;DE=DATA END
BE1E 7C        LD   A,H                  ;CHECK AGAIN
BE1F BA        CP   D
BE20 3803      JR   C,RAND2
BE22 D608      SUB  8                     ; -800H
BE24 67        LD   H,A
BE25 7E        RAND2: LD   A,(HL)         ;READ RND DATA (0-FFH)
BE26 69        LD   L,C
BE27 67        LD   H,A
BE28 CD03BF    CALL 0BF03H                ;L カケル H
BE2B 4C        LD   C,H                  ;SET C,H
;
BE2C E1        POP  HL
BE2D D1        POP  DE
;
BE2E 79        LD   A,C
BE2F 32FFBF    LD   (0BFFFH),A           ;TEST SET BFFFH
;
BE32 F1        POP  AF
BE33 C9        RET
;
BE34 0010      RAND0: DW   1000H          ;スタート DATA TOP
;
;
BE36          END

```



## リスト5-8b 実行プログラム

```

BE00 0E 10 F5 D5 E5 2A 34 BE 11 00 18 7C BA 38 03 21
BE10 00 10 23 22 34 BE ED 5F D5 5F 16 00 19 D1 7C BA
BE20 38 03 D6 08 67 7E 69 67 CD 03 BF 4C E1 D1 79 32
BE30 FF BF F1 C9 02 10 BF F1 C9 00 00 00 00 00 00 00


```

までの乱数になります。

かけ算は前使ったプログラムをそのまま使いますが、ランダムデータはどうしましょう。

ROMの値を適当に見てみる事にしましょう。リスト5-8がそのプログラムで、ランダムデータとして、ROMの1000Hから1800Hまでを参照しています。オールラムモードのゲームでは乱数データを作るのがよいと思います。

使い方は、Cレジスタの値に、乱数の上限+1の値を入れて、BE02をコールするのですが、テストする時にはBE01に値を入れて、

h] GBE00 

とすれば、BFFF番地に乱数が入ります。

もちろん、このプログラムは、リスト5-7を必要とします。

なお、リスト5-8のBE16番地からの命令に、

LD A, R

とありますが、これはAレジスタにリフレッシュカウンタの値を読み込む命令です。リフレッシュカウンタの値は0から7FHまでの範囲で高速に移動していますから、この範囲の乱数として使えます。しかし、ゲーム中で使う際に乱数の片寄りが起こる時がありますので、リフレッシュカウンタのみの使用はあまりおすすめできません。

なお、このプログラムを実際に使われる時には、キー入力待ちのループ部分にコールしてランダムデータの位置をかきまぜます。

## 5-12

### 周辺装置とお話する

ディスクドライブや、モデム、カプラなど外部とのデータのやりとりや、メモリバンクの切り替え、モードの設定などPC-88の機能を設定するのに、CPUに、IN・OUT命令を実行させて行います。

これらは、BASICの、INP関数およびOUT命令に対応します。

IN、OUTの使い方については、第2章I/Oの実戦的活用で、キースキャンや、ディスクコントロール、画面モードコントロールなど重要な部分について説明しています。

ここでは、確認の意味でIN・OUT命令の書式を掲載しておきます。

#### ●IN命令

PC-88は1バイト(256)個の入出力ポートを持っています。そこでこのポート番号を具体的に



指示する命令が、

`IN A, n` (nは1バイトの数)

です。ポートからのデータはAレジスタに入ります。これに対し、ポート番号をCレジスタに入れておいて、読み出す時には、

`IN r, (C)` (rはAからLまでのレジスタ)

とします。こちらは、Aレジスタでも他のBレジスタやCレジスタでもかまいません。実際にプログラムする時に、nやrは具体的に書きます。また、アセンブラを使う時はカッコやカンマを忘れないで下さい。

### ●OUT命令

IN命令と同様ですが、書く順が異なります。

`OUT n, A` (nは1バイト数)

`OUT (C), r` (rはAからLまでのレジスタ)

これらのマシンコードは付録C表にあります。

表中には、INIや、INIRなどブロック入出力と呼ばれる命令がありますが、ここでは省略します。

## 5-13

### その他の命令はあまり気にせず...

Z80の命令にはこの他、ビット操作命令、CPUコントロール命令、リスタート命令があります。これらの命令は必要な時に説明したいと思います。

一つの作業をするのでもプログラムは何通りかあるもので、たとえば、ビット操作も他の方法で間に合います。少しビット操作命令にふれておきましょう。

#### (i) BIT命令

あるビットが0か1かを調べる。レジスタの中味は変化しないが、ビットが0ならばゼロフラグがON、つまり1になる。例、`BIT 0, A` (Aレジスタの第0ビットが0か1か調べる)

#### (ii) SET命令

あるビットを1にする。たとえば、Aレジスタの第1ビットをセットしたければ、

`SET 1, A`

としますが、この命令は、

`OR 2`

で代用できます。その理由は、2は2進数で書くと、00000010で、これとAをORすることにより第1ビットが1になります。

#### (iii) RES命令

あるビットを0にする。たとえば、Aレジスタの第2ビットをリセットしたければ、

`RES 2, A`

としますが、この命令は、



## AND 0FBH

で代用できます。その理由は、FBHは2進数で書くと、11111011で、これとAをANDすることにより、第2ビットが0になります。

ANDやORを使うと、二つ以上のビットをセットしたり、リセットしたりできますので、BIT操作命令はあまり使いません。ただし、キースキャンにはBIT命令が便利です。

最後に、CPUコントロール命令のうちの“NOP”に注目して下さい。NOPとはノーオペレーションのことで、マシンコードは“00”です。つまり、Z80は“00”のコードを受け取ると何もしないで次へ行きます。

一見無意味なように見えますが、ハンドアセンブルには重要です。

たとえば、リスト5-7のかけ算ルーチンを使いたい時、CALL BF03 では、ちょっとはんばな数字ですね。そこで、BF00からBF02までを“00”にしてしまうと、CALL BF00 でも平気になります。

ハンドアセンブルでは、モニタで編集するので、サブルーチンの先頭をきりの良い数にした方が見やすいでしょう。そこで、プログラムの余りの部分が出ますが、そこを0で埋めるとよいでしょう。また、命令を削除したい時も、00で消してしまえば、他の部を書き替えたり、打ち込み直したりしないで済みます。

以上、Z80の使い方をごく大ざっぱに説明いたしましたが、サンプルプログラムなどをバンバン打ち込んで、まずやってみるのが第一です。プログラムが暴走した所で、パソコンが壊れる事はまずありませんよ。



究極の1チップパソコン



# マシン語少年物語

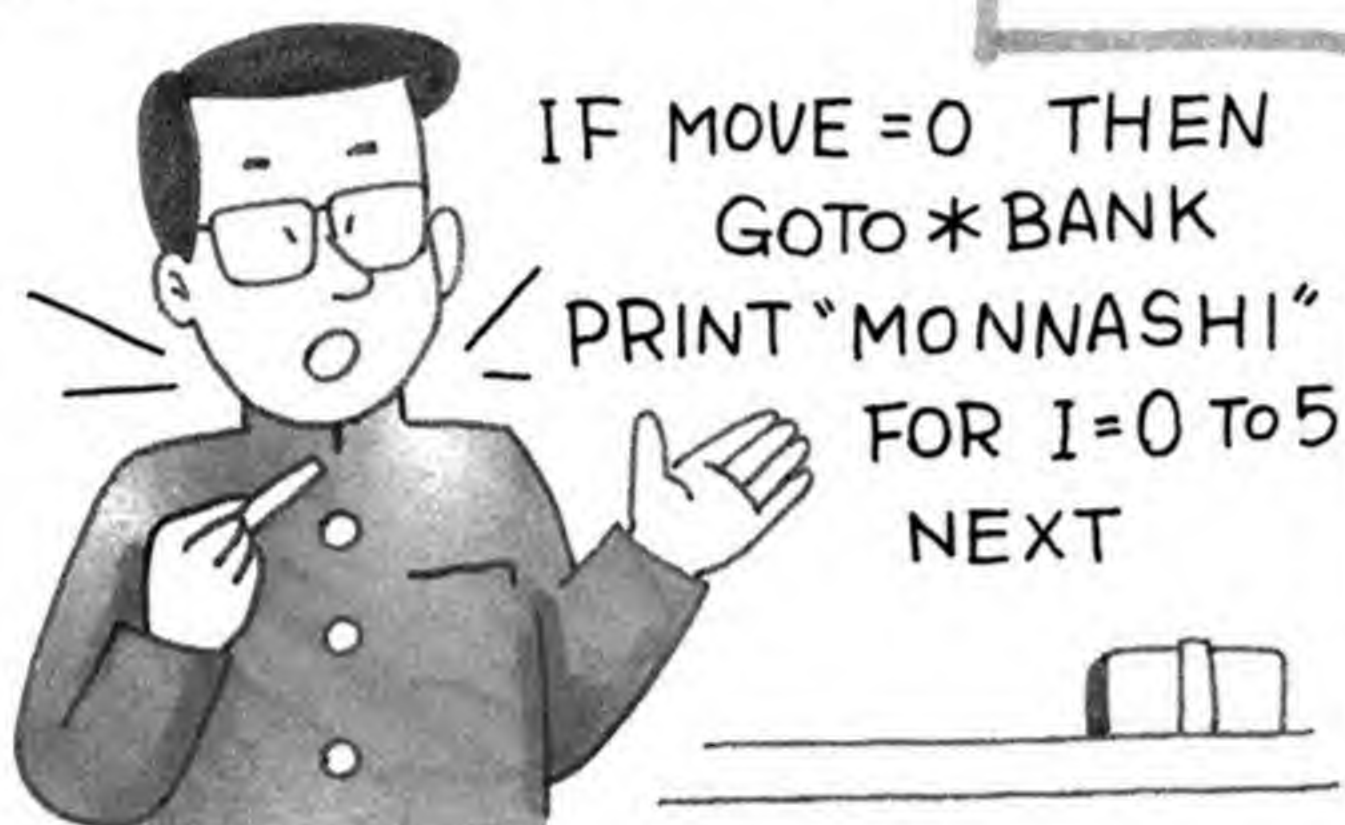
1. 生まれてまもなくマシン語しゃべり…



2. 小・中でコンパラを使い…



3. 高校でBASICをマスターし…



4. 20を過ぎ、ついに日本語をマスター。普通の人になった。





第 6 章

6

# マシン語ゲームを作ろう

6-1 まず、どんなゲームを作るか決める

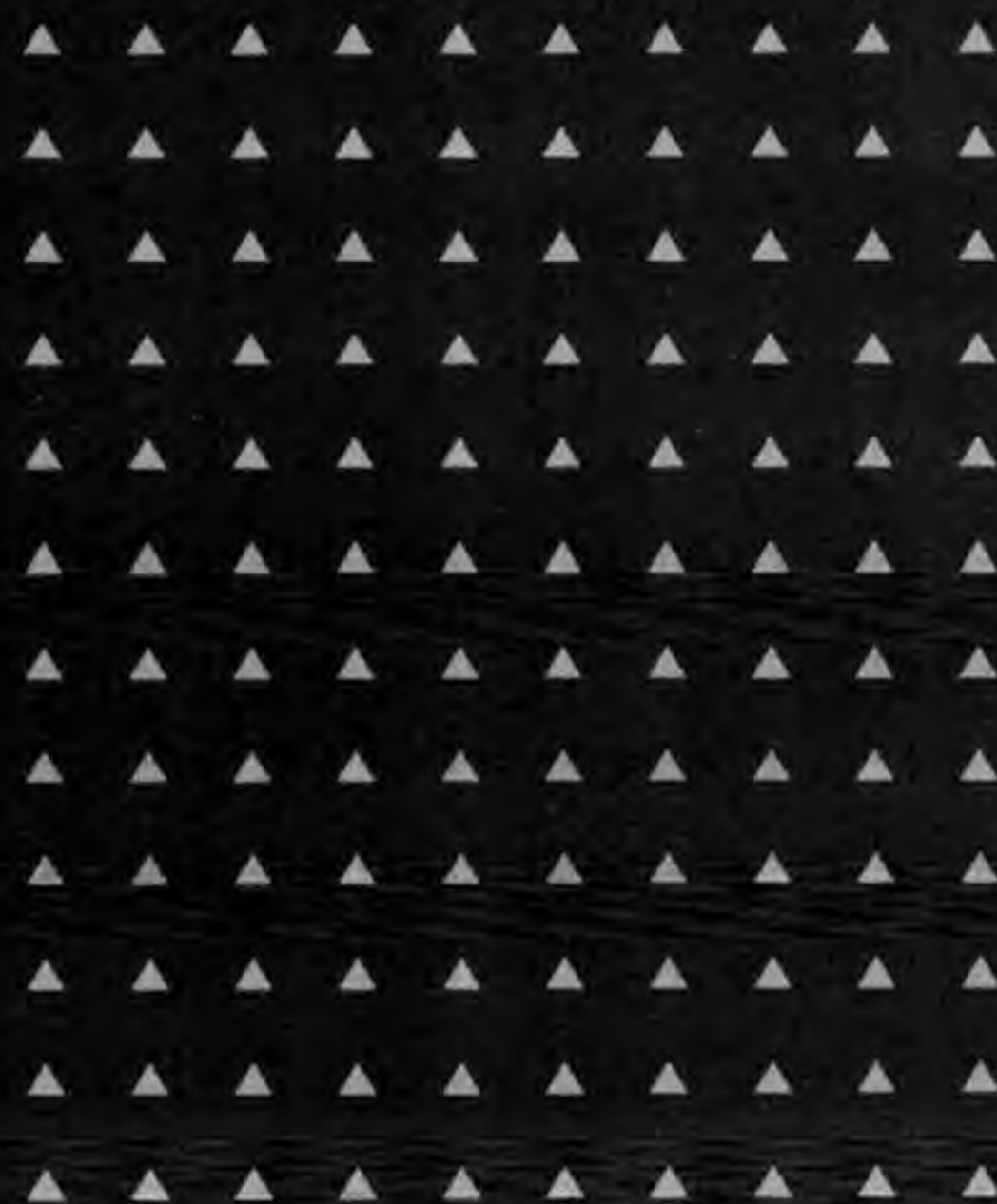
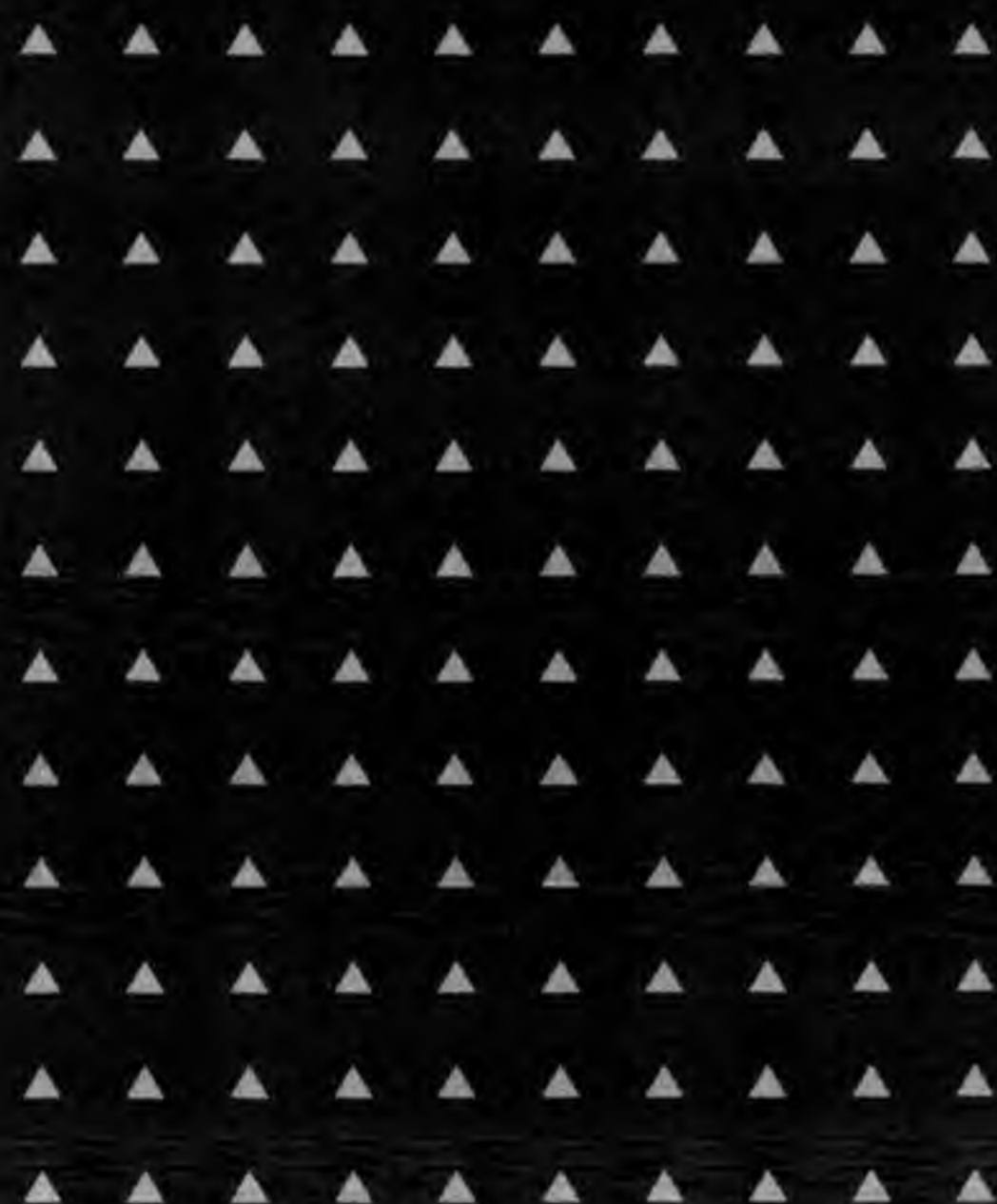
6-2 画面構成やフィールドを考える

6-3 ワークエリアを作りながら動かし方を考える

6-4 フィールドを作り、キャラクターを動かす

6-5 スクロール画面を作るには

6-6 終りに





## 第6章 マシン語ゲームを作ろう

### 6-1

#### まず、どんなゲームを作るか決める。

市販のゲームは、急速にその内容が大きくなってきました。ルールが複雑になればそれだけ作りにくくなるのは当然ですが、簡単なプログラムの積み重ねで、大きなゲームも成り立っているのです。

ゲームは複雑ならばよいと言うものでもなく、シナリオや、状況の設定次第でいくらかでも良い作品が生まれると思います。

私の場合、ゲームのアイデアは、小説やマンガによる所が多く、たとえば、“ある辺境の星で不思議な生物が発見され、調査のために探険隊が派遣される。彼等は危険を冒して卵を取り、本部へと送る。こうして何人もの犠牲により送られた卵は、実は大変な珍味で政府のオエライサンがひそかに食べてしまう。そうとも知らずに探険隊は今日もせっせと卵取り……”と、言うような話しから、卵→幼虫→サナギ→成虫と変化する生物の卵を取るゲームができました。成虫は“針”を撃って攻撃して来ます。さらに成虫は卵を4コ生み終えたと、“幽霊”となって攻撃します。

卵をある数取ると、一面クリア。3人死ぬか、生物を全滅させるとゲームオーバーです。成虫が多すぎると負けてしまうので、うまく成虫の数をコントロールする所がミソ。また、グラフィックは普通のグラフィックと、テキストのセミグラフィックを重ね合せ、森の中にいる状況を作ってみました(“ジャングル・ショック”)。

“沈む船を救うネズミ”と言う一コママンガがありました。船が危くなるとネズミが真っ先に逃げ出すのだそうですが、このネズミ君は船に開いた“穴”に自分の体を押し込んで水が入るのを防いでいます。

そこで、“ネズミが船底をかじり穴を開ける→水が入ってくる→水をポンプで汲み出す→ネズミをつかまえて、そいつを穴につめ込み補修する”と言うシナリオを作ってみました。ちょっと物足



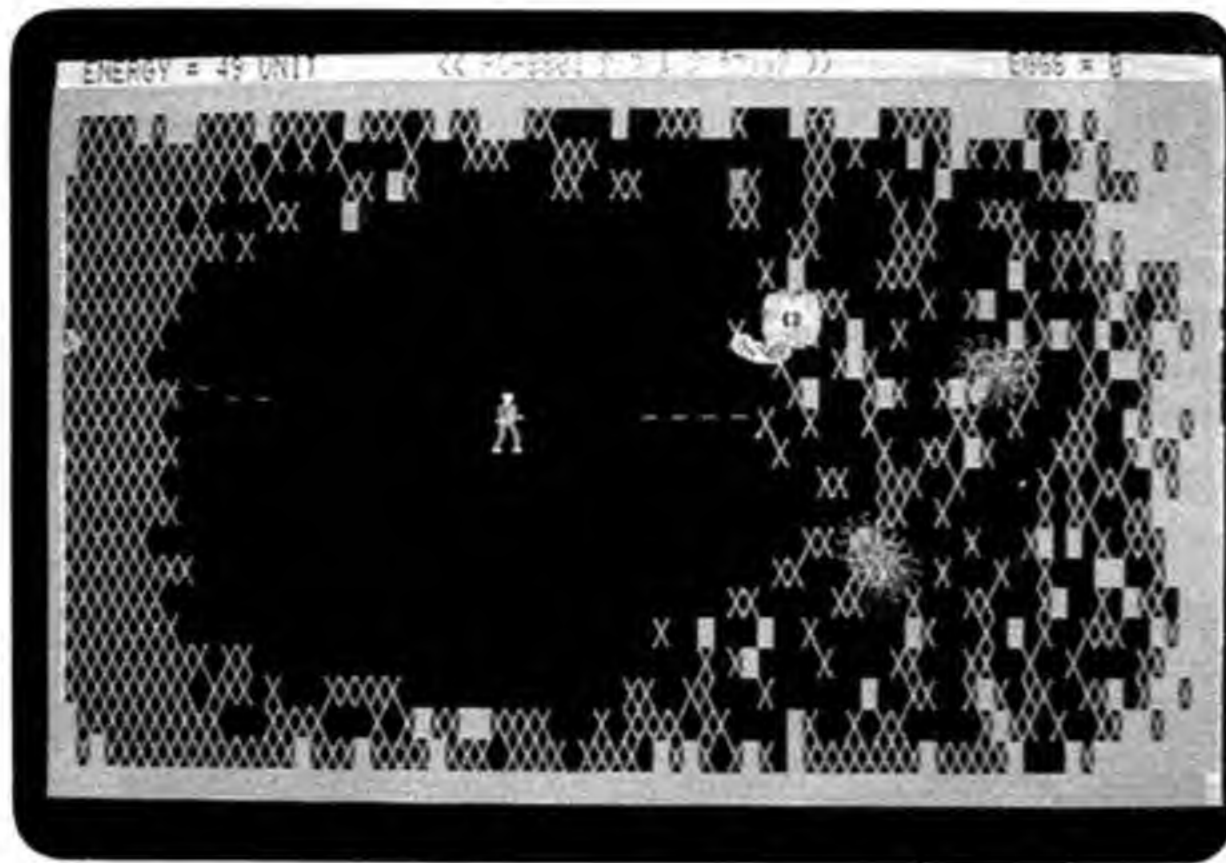


写真1 ジャングル・ショック

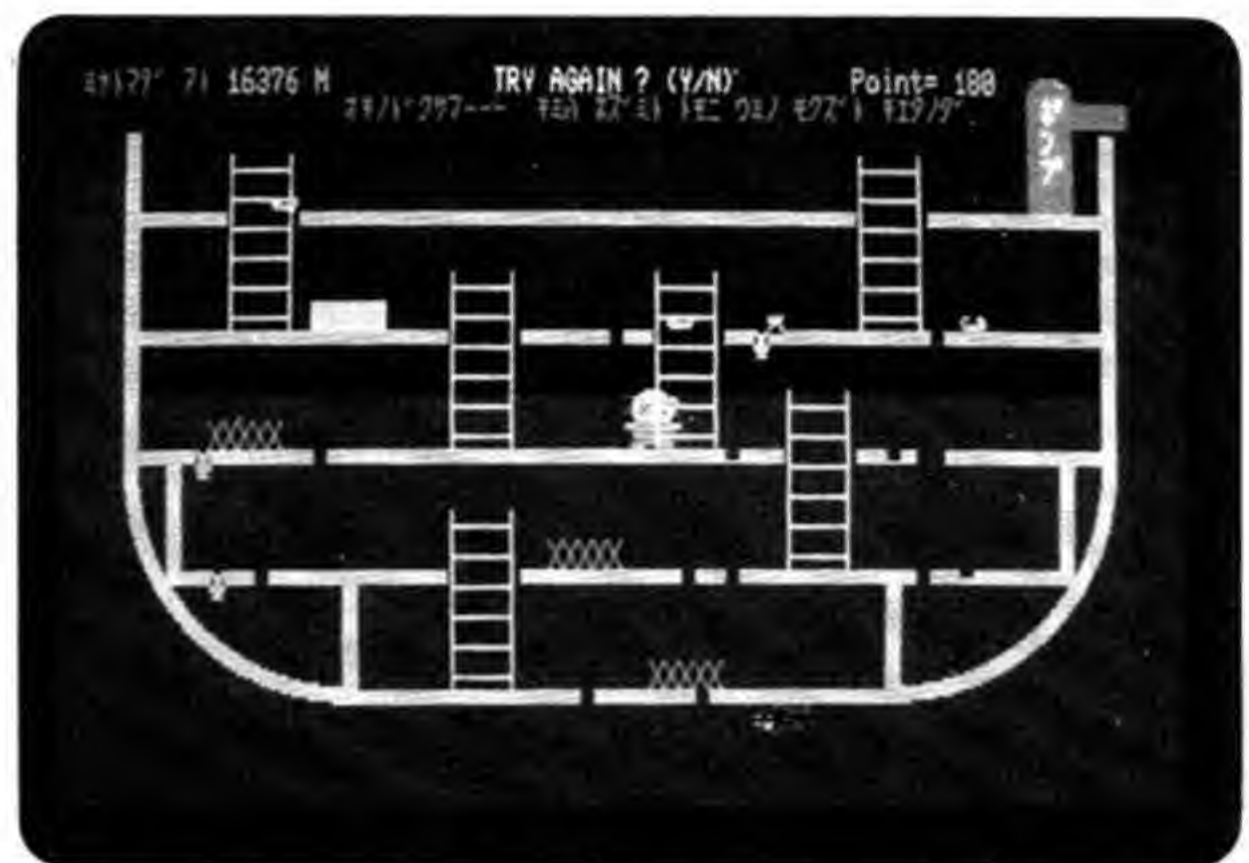


写真2 ポンコツ船サバイバル

りないので、敵に“ノコギリザメ”を加え、味方に“ネコ”を加えました（“ポンコツ船サバイバル”）。変ったアイデアのゲームだと思います。

## 6-2

### 画面構成やフィールドを考える

シナリオができ上がると、自然に画面のイメージができて来ますが、実際のゲーム画面を作るにはメインの画面の他に表示しなければならないものがいくつかでて来ます。たとえば“エプシロン3”の場合では3Dスクリーンの他に、マップ用のサブスクリーン、アイテム用コンソール、三つのエネルギーゲージにポイント、などです。

あまりゴチャゴチャしても見にくくなってしまうますが、必要な物は表示しなければなりません。表示しきれない時は、画面切り替えなどを考えねばなりません。

次に活動する場所をプログラムでわかるように設定しなければなりません。その必要がない場合もありますが、一般には行ける場所、行けない場所が出て来ます。

いくら画面に壁の絵を書いたからと言って、人間が絵を見るようにはコンピュータでは判断できません。

そこで、どこに何があるのかをメモリ上にデータで示し、プログラムでそれを見ながら動きを決めるのです。このデータ群を“仮想空間”と呼んだりしますが、多くのゲームでは、この仮想空間の様子を画面に映しているのです。私の場合、エプシロン3も GAMMA5 も、仮想空間のデータの使い方は同じで、上から見た絵にするか、3D的な絵にするかはプログラム次第と言えます。

仮想空間のデータの設定の方法は、人により様々ですが、具体的なものを後で紹介いたします。

これらのデータもシナリオ同様、始めからあまり複雑なものを手がけるのは大変です。

さて、今回は各種ルーチンの説明と実践のために、“追いかっこゲーム”を考えてみました。

ルールはとても単純、単純すぎてオモシロさは？という感じですが、学習用ですので許して下さい。

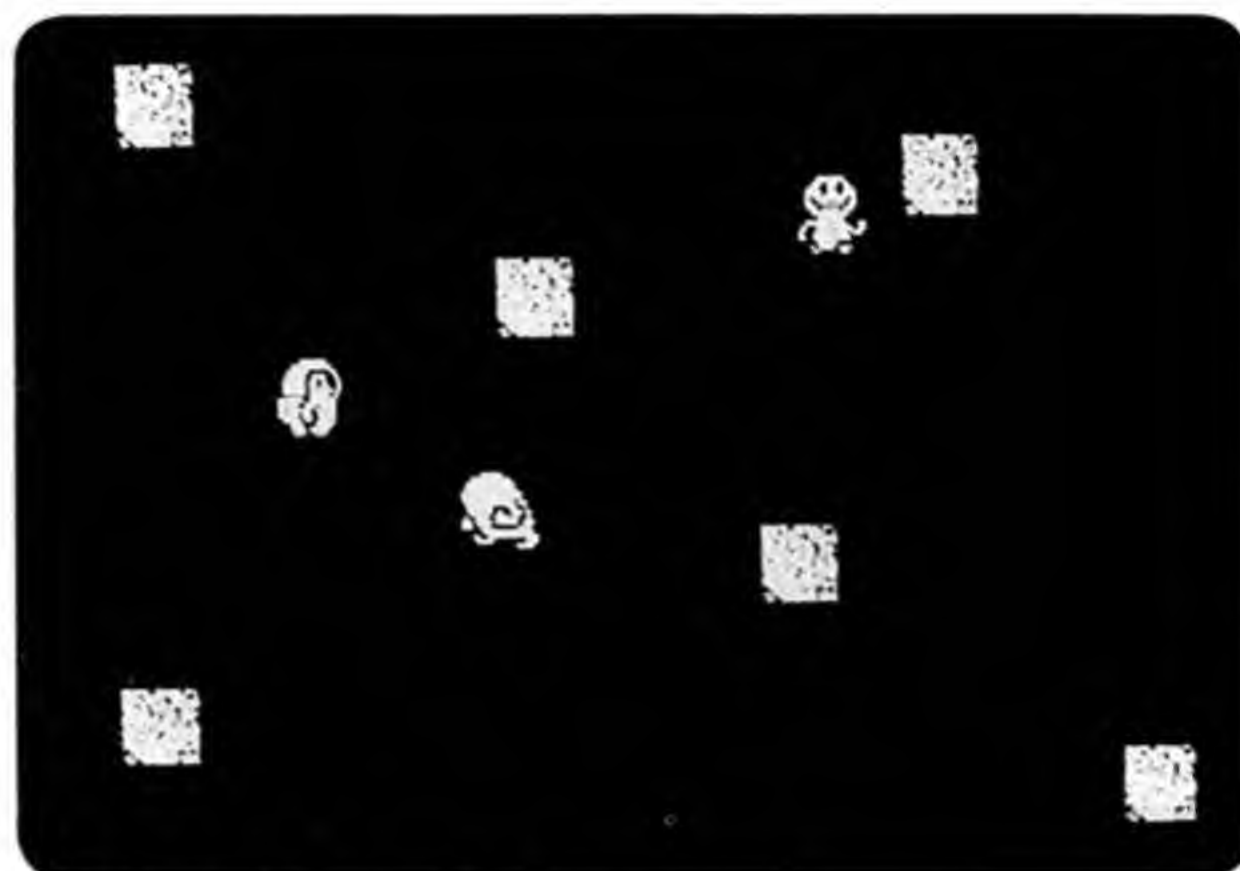
内容は2人の敵を捕えるものです。1人はランダムで動き、時々ワープします。1人はあなたを追いかけて来ます。敵の正面があなたに触れたらあなたの負け、あなたが敵の正面以外の所に



触れたら勝ちとします。

また参考にデータで移動する敵も作ってみました。ですから、これからの実験には、四つのキャラクターが必要で、前後左右2パターンずつ、それぞれに8パターン作る必要があります。

キャラクターの大きさは、横4バイト、縦16ラインにしましたので、前の“アストロノートと宇宙人”のデータを使うか、キャラクターツールで自作して下さい。



▲追いかっこゲーム

## 6-3

### ワークエリアを作りながら動かし方を考える

キャラクターを動かすには、何と言ってもまずその位置が決まらなければなりません。BASICですと“LOCATE”文などがあり、XY座標でその位置を決める事ができます。グラフィックを表示するのにも、同じ様にXYが使えると便利そうですね。

またキャラクターの形を決めるためには、“向き”や、アニメーションさせるための2パターンのうちどちらを使うか決めなければなりません。

第6-1表 ゲーム用ワークエリアの作成

マイキャラクターを動かす (&HE 000 番地より)

E 0 0 0	やられたかどうかのフラグ(ゲーム中：1, やられた：0)
E 0 0 1	X座標(0～76)
E 0 0 2	Y座標(0～46)
E 0 0 3	方向    下向き：0    右向き：1 上向き：2    左向き：3
E 0 0 4	ポーズ(0か1)
E 0 0 5	2バイト画面アドレス 下位
E 0 0 6	2バイト画面アドレス 上位
E 0 0 7	タイマー1(0からTまで変化) 0の時キャラクター移動
E 0 0 8	タイマー2(T) タイマーの最大値 値が大きい程動きが遅い



BASICでプログラムを作る時には、これらの要素は英数字で指定した“変数”に入れて置くのが普通です。マシン語では、レジスタが変数の役割をしていますが、わずかのレジスタでは沢山の変数処理できるはずがありません。

そこで、プログラマが設定する“変数”は、ある**特定のメモリ**に入れておき、Z80のレジスタは計算など、プログラムの処理にのみ使います。実はBASICでも変数名が指定されると、あるメモリをその変数に割り当てています。

これらのメモリがある所をワークエリアと言い、プログラム作りに当って、必要な変数を考え、ワークエリアを作ります。**第6-1表**が具体的に自分の移動に必要な変数を決めたもので、**第6-2表**は他のキャラクタも含めたワークエリア全体の様子です。

ワークエリアは、プログラム中でも必要となる時がありますが、それはその都度プログラムの近くに作るかワークエリアに設定します。

## 6-4

### フィールドを作り、キャラクタを動かす

#### (I) 動かすために必要なサブルーチンを準備する

まず、もっとも基本的なルーチンから考えてみましょう。キャラクタを移動させるには、画面にあるキャラクタを消し、新しい場所に新しいキャラクタをプットするのですから、クリアルーチンと、プットルーチンはまず必要です。

**リスト6-1**はキャラクタのプットルーチンと、クリアルーチンです。第5章のルーチンを使っていますが、前のルーチンが、&HBFF0からワークエリアにしていたのに対し、**第6-2表**によるワークエリアを使っています。

このサブルーチンの使い方は、HLレジスタの中に&HE000, &HE010, &HE020, &HE030 のどれかを入れてコールします。

たとえばHLレジスタに&HE000を入れて、

第6-2表 ワークエリア割り付け

	0	1	2	3	4	5	6	7	8	9	A
E 000(マイキャラクタ)										●	
E 010(データ移動敵)										●	●
E 020(追いかける敵)										●	●
E 030(ランダム移動敵)										●	

部分は第6-1表に同じ

E 009→倒した敵の数

E 019,E 01A→移動データのアドレス(2バイト)

E 029,E 039→ランダム変化の割合

E 02A→追いかける割合



リスト6-1 キャラクタのプットルーチン, クリアルーチン

```

;
; CHR PUT                                IN HL, WARK TOP
;
;
;      ORG 0BF00H
;
BF00 F3      CPUT: DI                      ;ワリミ タメ
BF01 3AC2E6   LD  A,(0E6C2H)
BF04 F602     OR  2
BF06 D331     OUT (31H),A                  ;ALL RAM MODE
;
BF08 CD50BE   CALL CALCG                  ;G ADDRESS ケイサン
BF0B CD80BE   CALL CALCC                  ;CHR ADDRESS ケイサン
BF0E 0604     LD  B,4                      ;4 BYTES
BF10 0E10     LD  C,10H                   ;16 LINE
;
BF12 C5      CPUTL: PUSH BC                ;LOOP START
;
BF13 D5       PUSH DE
BF14 48       LD  C,B                      ;BC=B (X)
BF15 0600     LD  B,0
BF17 C5       PUSH BC
BF18 D35C     OUT (5CH),A                  ;SELECT BLUE RAM
BF1A EDB0     LDIR                        ;(DE)=(HL) フロック テンソウ
BF1C D35F     OUT (5FH),A                  ;SELECT MAIN RAM
BF1E C1       POP  BC                      ;X SIZE モトス
BF1F D1       POP  DE                      ;G サビョウ モトス
;
BF20 D5       PUSH DE                      ;オナシ コト ノ クリカエシ
BF21 C5       PUSH BC
BF22 D35D     OUT (5DH),A                  ;SELECT RED RAM
BF24 EDB0     LDIR
BF26 D35F     OUT (5FH),A                  ;SELECT MAIN RAM
BF28 C1       POP  BC
BF29 D1       POP  DE
;
BF2A D5       PUSH DE                      ;オナシ コト ノ クリカエシ
BF2B D35E     OUT (5EH),A                  ;SELECT GREEN RAM
BF2D EDB0     LDIR
BF2F D35F     OUT (5FH),A                  ;SELECT MAIN RAM
BF31 D1       POP  DE
;
BF32 E5       PUSH HL
BF33 215000   LD  HL,50H
BF36 19       ADD HL,DE                    ;NEXT LINE
BF37 EB       EX  DE,HL
BF38 E1       POP  HL
;
BF39 C1       POP  BC
BF3A 0D       DEC  C
BF3B 20D5     JR   NZ,CPUTL
;
BF3D 3AC2E6   LD  A,(0E6C2H)                ;BASIC ROM MODE
BF40 D331     OUT (31H),A
;
BF42 FB       EI                          ;ワリミ OK !
BF43 C9       RET
;

```



```

; ALU CLS
;
; IN HL, WARK TOP
;
; ORG 0BF50H
;
BF50 F3      CRESET: DI                      ;ワリコミ タメ
;
BF51 3AC2E6   LD  A, (0E6C2H)
BF54 F602     OR   2
BF56 D331     OUT (31H), A                  ;ALL RAM MODE
;
BF58 23       INC  HL
BF59 23       INC  HL
BF5A 23       INC  HL
BF5B 23       INC  HL
BF5C 23       INC  HL
BF5D 5E       LD   E, (HL)
BF5E 23       INC  HL
BF5F 56       LD   D, (HL)
BF60 EB       EX   DE, HL                  ;HL=G ADDRESS
BF61 0604     LD   B, 4                    ;X 4 BYTES
BF63 0E10     LD   C, 10H                 ;Y 16 LINES
;
BF65 DB32     IN   A, (32H)
BF67 F5       PUSH AF
BF68 F640     OR   40H                    ;KAKUTHIOO MODE
BF6A D332     OUT (32H), A
;
BF6C 3E80     LD   A, 80H
BF6E D335     OUT (35H), A                ;KAKIKOMI
BF70 AF       XOR  A
BF71 D334     OUT (34H), A                ;BIT RESET MODE
;
BF73 C5       CRESL1: PUSH BC
BF74 E5       PUSH HL
BF75 3EFF     LD   A, 0FFH                ;FULL BIT RESET
BF77 77       CRESL2: LD   (HL), A
BF78 23       INC  HL
BF79 10FC     DJNZ CRESL2
BF7B E1       POP  HL
BF7C 015000   LD   BC, 50H
BF7F 09       ADD  HL, BC                 ;NEXT LINE
BF80 C1       POP  BC
BF81 0D       DEC  C
BF82 20EF     JR   NZ, CRESL1
;
BF84 AF       XOR  A
BF85 D335     OUT (35H), A                ;MAIN RAM
BF87 F1       POP  AF                    ;NORMAL MODE
BF88 D332     OUT (32H), A
BF8A D35F     OUT (5FH), A                ;RAM
;
BF8C 3AC2E6   LD   A, (0E6C2H)
BF8F D331     OUT (31H), A
BF91 FB       EI
BF92 C9       RET
;
;

```



## CALL 0BF50H

を実行すると、E005番地とE006番地に入っている画面番地の所から、横4バイト、縦16ライン分を消してしまいます。

&HBF00番地からのプットルーチンでは、xy座標から画面アドレス(2バイト)を求めるルーチンと、方向と、ポーズによってキャラクタのデータアドレスを求めるルーチンをコールしてから、ブロック転送によるプットを行っています。

プットするデータの先頭アドレスと、プット先のグラフィックメモリの先頭アドレスがわかれば、前のプットルーチンがそのまま使えるわけです。

そこで、リスト6-2の&HBE50から始まっているのが、XY座標から画面用メモリのアドレスを求めるプログラムです。

注意していただきたいのは、BASICのLINE文等で使われているx,yは、xが0から639まで、yが0から199までですが、ここで設定したXは0から79まで、Yは0から49までです。

と言っても、決して移動範囲をせばめたわけではありません。画面の番地を考えるとわかると思いますが、横は1ライン80バイトでできています(1バイトにつき8ドットの点がうてるので、 $80 \times 8 = 640$ ドットになるのですね)。

つまり選べる所が横640ではなく、80のうちのどこかになり、一つ隣にキャラクタを移すと8ドット分動く事になります。従ってゲームのX座標は0から79に設定するのがよいでしょう(もちろん

リスト6-2 XY座標から画面用メモリのアドレス計算

```

;
;
; G ADDRESS FROM X,Y      IN HL,WARK TOP  OUT DE,G ADDRESS
;
;
;          ORG  0BE50H
;
BE50 E5      CALCG: PUSH HL
BE51 23      INC  HL
BE52 5E      LD   E,(HL)      ;READ X
BE53 23      INC  HL
BE54 56      LD   D,(HL)      ;READ Y
BE55 E5      PUSH HL
BE56 D5      PUSH DE
BE57 62      LD   H,D          ;H=Y
BE58 CB24    SLA  H
BE5A CB24    SLA  H            ;Y*4=LINE
BE5C 2E50    LD   L,80         ;1 LINE 80 BYTES
BE5E CD00BE  CALL 0BE00H       ;L*H
BE61 1100C0  LD   DE,0C000H    ;SCREEN TOP ADDRESS
BE64 19      ADD  HL,DE        ;HL=HL+DE
BE65 D1      POP  DE
BE66 1600    LD   D,0          ;DE=X
BE68 19      ADD  HL,DE        ;GET G ADDRESS !
BE69 EB      EX   DE,HL
BE6A E1      POP  HL
BE6B 23      INC  HL
BE6C 23      INC  HL
BE6D 23      INC  HL          ;SKIP 3 BYTES
BE6E 73      LD   (HL),E       ;SET ADDRESS LOW
BE6F 23      INC  HL
BE70 72      LD   (HL),D       ;SET ADDRESS HIGH
BE71 E1      POP  HL
BE72 C9      RET
;

```



```

; CHR ADDRESS FROM ホウコウ,ハ°ターン スイッチ IN HL,WORK TOP OUT HL,C AD
;
;
; ORG 0BE80H
;
BE80 C5 CALCC: PUSH BC
BE81 D5      PUSH DE
BE82 7D      LD A,L
BE83 A7      AND A ;A=0 ラ シラヘ"ル
BE84 2005    JR NZ,CALCC1
BE86 010060  LD BC,6000H ;ASTRO DATA TOP
BE89 1815    JR CALCC4
;
BE8B FE10 CALCC1:CP 10H
BE8D 2005    JR NZ,CALCC2 ;IF A<>10H THEN
BE8F 010066  LD BC,6600H ;ウチウシ"ン 1
BE92 180C    JR CALCC4
;
BE94 FE20 CALCC2:CP 20H
BE96 2005    JR NZ,CALCC3 ;IF A<>20H THEN
BE98 01006C  LD BC,6C00H ;ウチウシ"ン 2
BE9B 1803    JR CALCC4
;
BE9D 010072 CALCC3:LD BC,7200H ;ウチウシ"ン 3
;
BEA0 23 CALCC4:INC HL ;BASE ADDRESS IN BC
BEA1 23      INC HL
BEA2 23      INC HL
BEA3 56      LD D,(HL) ;SKIP 3 BYTES
BEA4 23      INC HL ;READ DIRECT.
BEA5 5E      LD E,(HL)
BEA6 15      DEC D ;READ POUSE
BEA7 200D    JR NZ,CACCB1
BEA9 1D      DEC E
BEAA 2005    JR NZ,CACCR1
BEAC 210000  LD HL,0 ;ミキ" 1
BEAF 1830    JR CALCC5
;
BEB1 218001 CACCR1:LD HL,180H ;ミキ" 2
BEB4 182B    JR CALCC5
;
BEB6 15 CACCB1:DEC D
BEB7 200D    JR NZ,CACCL1
BEB9 1D      DEC E
BEBA 2005    JR NZ,CACCB2
BEBD 218004  LD HL,480H ;ウシ" 1
BEBC 1820    JR CALCC5
;
BEC1 214005 CACCB2:LD HL,540H ;ウシ" 2
BEC4 181B    JR CALCC5
;
BEC6 15 CACCL1:DEC D
BEC7 200D    JR NZ,CACCF1
BEC9 1D      DEC E
BECA 2005    JR NZ,CACCL2
BECC 21C000  LD HL,0C0H ;ヒタ"リ 1
BECF 1810    JR CALCC5
;
BED1 214002 CACCL2:LD HL,240H ;ヒタ"リ 2
BED4 180B    JR CALCC5
;
BED6 1D CACCF1:DEC E
BED7 2005    JR NZ,CACCF2
BED9 210003  LD HL,300H ;ミ 1
BEDC 1803    JR CALCC5
;
BEDE 21C003 CACCF2:LD HL,3C0H ;ミ 2
;
BEE1 09 CALCC5:ADD HL,BC ;HL=HL+BC GET C ADDRESS !
BEE2 D1      POP DE
BEE3 C1      POP BC
BEE4 C9      RET

```



ん他の値にしてもよい)。

Xが一つにつき8ドットずつですから、Yは4ラインずつの移動にすると長さの比が合います。だから、Y座標は0から49までと言う事になります。

グラフィックRAMの先頭アドレス(画面左上)は&HC000ですから、X、Yから画面アドレスを求める式は次の様になります。

$$\text{画面番地} = \&\text{HC000} + Y \times 4 \times 80 + X$$

$Y \times 4 \times 80$ は、Y一つで4ラインですから、 $Y \times 4$ でライン数を求め、1ライン80バイトですから、これに80をかけてバイト数を求めているわけです。

ここで求めた画面アドレスは、ワークエリアに保存しておき、クリアする時に使っています。

さて、リスト6-2の&HBE80からはキャラクタの先頭番地を求めるプログラムです。このプログラムでは4種類各8パターン合計32パターンのデータが、&H6000番地から&H77FF番地までにあるものとして計算を行っています。

計算の手順は次の通りです。

1. 計算したい物のワークエリア先頭アドレスを受け取り、それを調べて4種類のどれかを判断し、キャラクタ先頭アドレスを決定する。
2. ワークエリアのデータによりキャラクタの方向を見る。方向は四通りあるので、その一つ一つについて場合わけし、ワークエリアのポーズデータから二通りあるうちの一つを決定、先頭キャラクタからの距離をHLレジスタに入れる。
3. 先頭キャラクタ番地と距離を足し合わせると、求めたい番地がHLレジスタに入る。

以上です。私の作ったキャラクタデータの順序が少し変でしたので、ちょっと長いプログラムになってしまいました。

さて、座標計算プログラムではかけ算ルーチンが必要です。また、ランダムルーチンも後で必要になりますから、これらをリスト6-3にのせておきます。前に全く同じものがありますが、ランダムルーチンのリフレッシュカウンタの値を足す所(&HBE3Aの"19")は、ない方がよいかもしれません(&HBE3Aを"00"にする)。実際に使ってみて、そんな感じを受けました。

## (II) 動き回るフィールドを作る

さて、キャラクタが画面上どこでも動き回れると言うのでは、ちょっと物足りません。と言っても様々な物体を画面上に置くのは、複雑になってしまいますから、障害物一種類のみを作ります。

そこで仮想空間の設定ですが、単純に考えて、XY座標一つにつき1バイトを割り当ててみましょう。メモリは&HD000から使うとすると、Xが80、yが50までありますから、 $80 \times 50 = 4000$ 、つまり4Kバイト必要です。これは画面のメモリに較べて4分の1(yが4ラインで一つだから)の大きさですが、工夫によってはもっともっと小さなメモリで済みます(第6-1図参照)。

この仮想空間の&HD000からのメモリの内容が、"00"ならば通過できる所、それ以外の数字ならばいけない所とします(一つのメモリにつき256通りの情報が入るのに、二種類にしか使わないのはもったいないですが……)。

障害物の大きさは適当でかまいませんが、ここではキャラクタの大きさと同じにしてみました。



キャラクタの大きさは、XYそれぞれ四つずつの大きさですから、 $4 \times 4 = 16$ で、仮想空間の16個

リスト6-3 かけ算ルーチン, ランダムルーチン

```

;
; L カケル H      IN H,L   OUT HL
;
;
;          ORG  0BE00H
;
BE00 F5      LKAKEH: PUSH AF
BE01 C5              PUSH BC
BE02 AF              XOR  A           ;A=0
BE03 0608        LD   B,8           ;8 カイ シフト
BE05 CB1D      KAKER1: RR  L         ;L ミキ" シフト
BE07 3001              JR  NC,KAKER2 ;ソノ ケタ カ" 0 ナラ シ" ャンフ°
BE09 84              ADD  A,H        ;1 ナラ A ニ H ラ タス
BE0A 1F      KAKER2: RRA           ;A ミキ" シフト
BE0B 10F8              DJNZ KAKER1
BE0D CB1D              RR  L         ;L ミキ" シフト
BE0F 67              LD   H,A       ;H ニ A ラ イレル
BE10 C1              POP  BC
BE11 F1              POP  AF
BE12 C9              RET

;
; 8 BIT ランダ" ム メイカー      IN C   OUT 0 カラ C-1
;          ランダ" ム デ" ター 1000H-1800H / ROM DATA
;
;          ORG  0BE20H
;
BE20 F5      RAND:  PUSH AF
BE21 D5              PUSH DE
BE22 E5              PUSH HL

;
BE23 2A4EBE        LD   HL,(RAND0)   ;READ DATA ADDRESS
BE26 110018        LD   DE,1800H     ;DATA END
BE29 7C              LD   A,H        ;LIMIT CHECK
BE2A BA              CP   D
BE2B 3803              JR   C,RAND1
BE2D 210010        LD   HL,1000H     ;DATA TOP
BE30 23      RAND1: INC  HL          ;DATA ADDRESS +1
BE31 224EBE        LD   (RAND0),HL
BE34 ED5F              LD   A,R      ;READ リフレッシュ レジ" スタ
BE36 D5              PUSH DE
BE37 5F              LD   E,A
BE38 1600              LD   D,0
BE3A 19              ADD  HL,DE       ;DATA ADDRESS +R (0-7FH RND)
BE3B D1              POP  DE         ;DE=DATA END
BE3C 7C              LD   A,H        ;CHECK AGAIN
BE3D BA              CP   D
BE3E 3803              JR   C,RAND2
BE40 D608              SUB  8         ;-800H
BE42 67              LD   H,A
BE43 7E      RAND2: LD   A,(HL)      ;READ RND DATA (0-FFH)
BE44 69              LD   L,C
BE45 67              LD   H,A
BE46 CD00BE        CALL 0BE00H       ;L カケル H
BE49 4C              LD   C,H        ;SET C,H

;
BE4A E1              POP  HL
BE4B D1              POP  DE
BE4C F1              POP  AF
BE4D C9              RET

;
BE4E 0010      RAND0: DW   1000H     ;スタート DATA TOP

```



の場所を0以外の数字で埋めなければなりません。


たとえば、 $X=0\sim3$ ,  $Y=0\sim3$ の所に障害物を置くとすると、&HD000~&HD003, &HD050~&HD053, &HD0A0~&HD0A3, &HD0F0~&HD0F3までに0以外の数字を入れる事になります。

この様な作業は手で行うのは面倒ですから、普通はBASICで簡単なツールを自作します。参考までにリスト6-4にサンプルプログラムをのせておきます。

このプログラムの使い方は、あらかじめ、


CLEAR, &HCFFF 

とした後、

MON 


h]FD000, DFFF, 0 

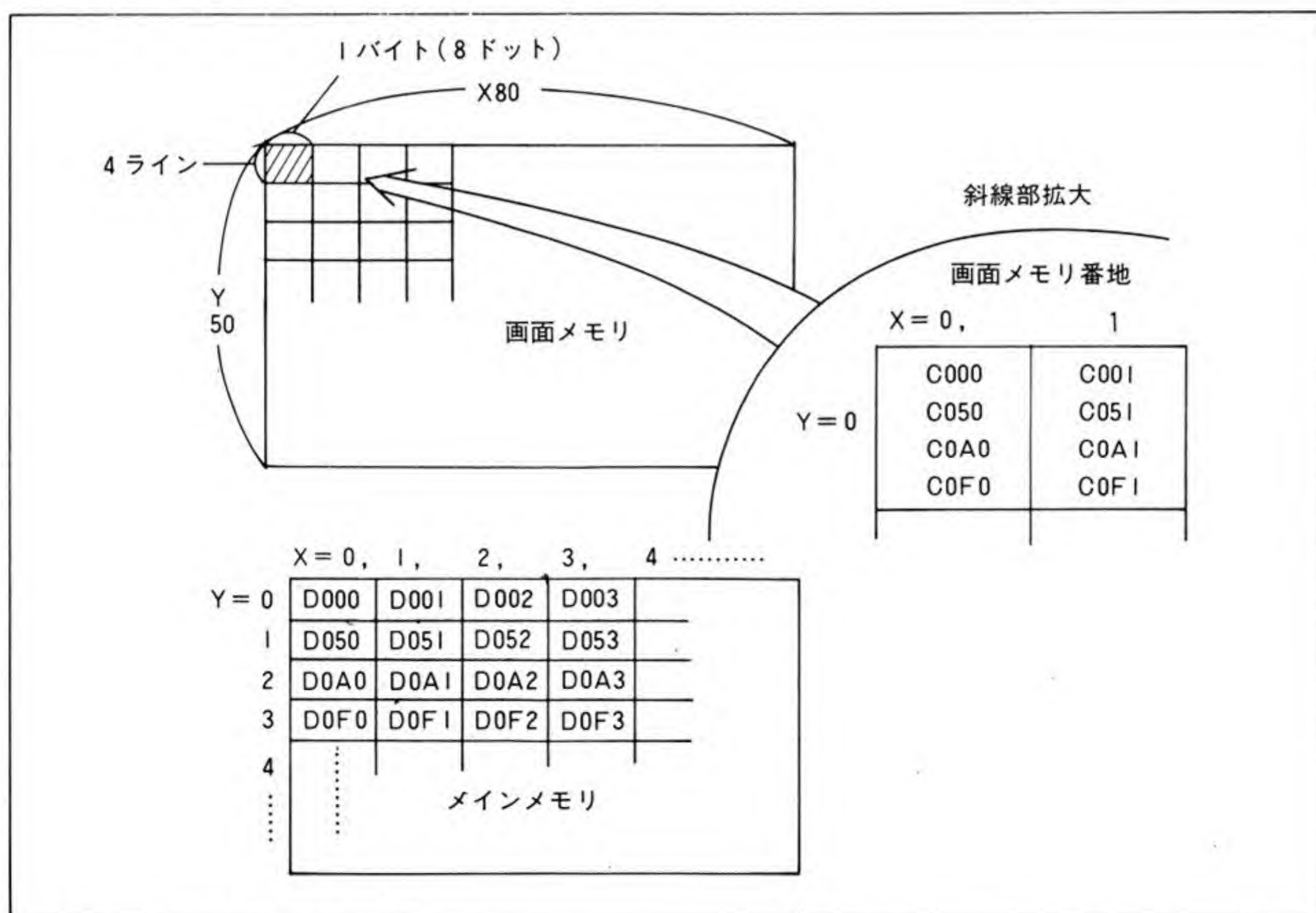
とやって、Fコマンドで、仮想空間メモリを全部0で埋めてしまいます。

そして、BASICプログラムをRUNさせると、画面左上にカーソルが出現します。このカーソルを移動させるのは、2, 4, 6, 8のキーを使います。希望の所で、5のキーを押すと、緑の四角が現れ、その場所に相当するメモリに数値が16個セットされます。

数は、先頭(左上に当る所)が"02"で、その他が"01"です。先頭だけ02にしたのは、障害物を表示するのに都合がよいからです。

作ったデータは、

BSAVE "ファイル名", &HD000, &H1000 



第6-1図 実画面と仮想空間



リスト6-4 フィールドメイカー

```

100 WIDTH 80,25:CONSOLE 0,25,0,1:X=0:Y=0:ADTOP=&HD000
110 LOCATE X*2,Y:A=ASC(INPUT$(1))
120 IF A=&H34 THEN IF X=0 THEN 110 ELSE X=X-1:GOTO 110
130 IF A=&H36 THEN IF X=38 THEN 110 ELSE X=X+1:GOTO 110
140 IF A=&H38 THEN IF Y=0 THEN 110 ELSE Y=Y-1:GOTO 110
150 IF A=&H32 THEN IF Y=23 THEN 110 ELSE Y=Y+1:GOTO 110
160 IF A<>&H35 THEN 110
170 LINE (X*16,Y*8)-STEP(32,16),4,BF
180 AD=ADTOP+X*2+Y*160:LOCATE 0,0:PRINT HEX$(AD);
190 FOR I=0 TO 3:FOR J=0 TO 3
200 POKE AD+J+I*80,1:NEXT:NEXT
210 POKE AD,2
220 GOTO 110

```

リスト6-5 障害物を描く

```

;
; DRAW ショウカ`イフ`ツ
;
;
;          ORG 0BCA0H
;
BCA0 2100D0  DGOMI: LD    HL,0D000H          ;DATA TOP
BCA3 1100C0          LD    DE,0C000H          ;G ADDRESS
BCA6 0E2E          LD    C,46
BCA8 0650  DGOMI1:LD    B,80
BCAA D5          PUSH DE
BCAB C5  DGOMI2:PUSH BC
BCAC 7E          LD    A,(HL)
BCAD FE02         CP    2
BCAF CCC3BC       CALL Z,DGOMIS
BCB2 23          INC    HL
BCB3 13          INC    DE
BCB4 C1          POP    BC
BCB5 10F4        DJNZ  DGOMI2
BCB7 D1          POP    DE
BCB8 E5          PUSH HL
BCB9 214001       LD    HL,140H              ;4 LINE
BCBC 19          ADD    HL,DE
BCBD EB          EX     DE,HL
BCBE E1          POP    HL
BCBF 0D          DEC    C
BCC0 20E6        JR     NZ,DGOMI1
BCC2 C9          RET
;
BCC3 D5  DGOMIS:PUSH DE
BCC4 E5          PUSH HL
BCC5 F3          DI
BCC6 210010       LD    HL,1000H            ;ROM DATA TOP
BCC9 CD0EBF       CALL 0BFOEH              ;DRAW ROM DATA GOMI
BCCC E1          POP    HL
BCCD D1          POP    DE
BCCE C9          RET
;
;

```



とやって、ディスクにセーブしておきましょう。

この作った障害物を画面に表示するプログラムが、リスト6-5です。仮想空間に"02"のデータがあったとき、そこに相当する画面位置に、4バイト、16ラインのキャラクタを置きます。キャラクタデータは&H1000番地からのROMの内容を持って来ているので、ゴチャゴチャの固りです。もし、自作のキャラクタを使われる時は、オールRAMモードに設定するなどの書き替えが必要です(第6-2図)。

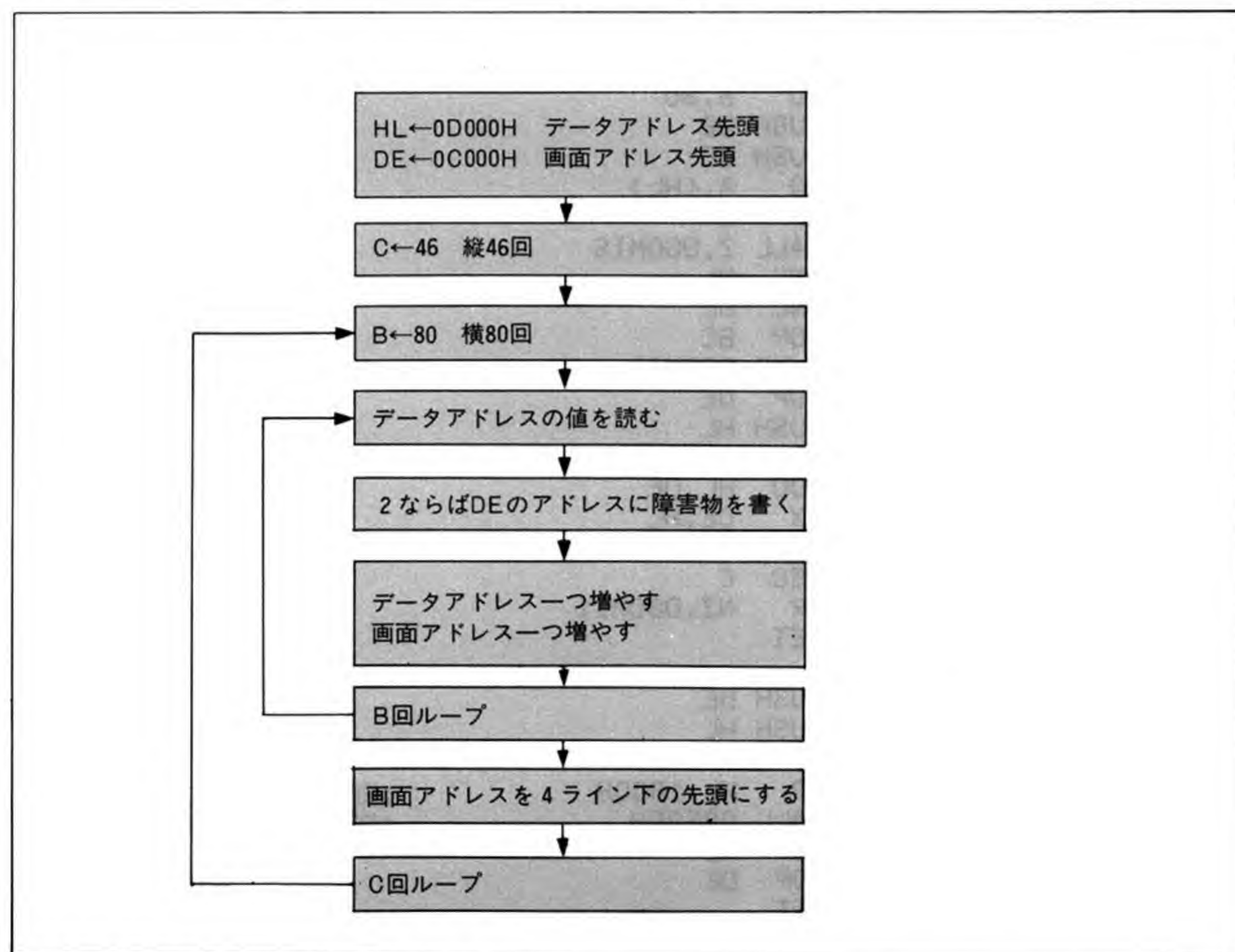
なお、プットルーチンはリスト6-1の途中からを使っていますので、テストする時にはこちらも忘れないで下さい。

### (III) “判定”する

自分のキャラクタが敵にぶつかったり、壁に当たったり、弾に当たってやられたりを判断するのは、ゲームプログラムではなくてはならない部分です。

これらの判定は、大きく分けて二つに分けられます。一つはマップ、つまり仮想空間のデータによる判定で、“落ちている物をひろう”とか、“壁で進めない”と言ったものです。

もう一つは動いている敵に当たったなどで、お互いのXY座標の比較による判定です。もっとも、やり方によってはどちらか一方にまとめられます(動いている敵も仮想空間中のデータにするなど)。



第6-2図 リスト6-5「障害物を描く」手順



ではまず、仮想空間のデータにより、進めるか進めないかの判定を行う方法を見てみましょう。

リスト6-6は、EレジスタにX座標、DレジスタにY座標の値を入れて、

CALL &HBDC0

とすると、キャラクタが障害物に重なる時はキャリーフラグを1に、重ならない時は0にして戻って来ます(第6-3図参照)。

キャラクタの大きさが、X四つ、Y四つ分ありますので、それぞれについてみていますが、これならば移動方向を考えずにチェックできます。

次に動いているもの同志の衝突の判定を考えてみましょう。

リスト6-6 仮想空間とXYとの判定

```

;
; FILED CHECK          IN E=X,D=Y
;                      OUT CY
;
;          ORG 0BDC0H
;
BDC0 C5      FCHECK: PUSH BC
BDC1 D5      PUSH DE
BDC2 0E04    LD C,4          ;LOOP COUNTER
BDC4 0604    LD B,4          ;LOOP COUNTER
BDC6 C5      FCHEC1: PUSH BC
BDC7 D5      PUSH DE
BDC8 CDE1BD FCHEC2: CALL GETWAD
BDCB A7      AND A
BDCC 200D    JR NZ,FCHEC3    ;A<>0 ショウカ"イフ"ツ
BDCE 1C      INC E
BDCF 10F7    DJNZ FCHEC2
BDD1 D1      POP DE
BDD2 C1      POP BC
BDD3 14      INC D
BDD4 0D      DEC C
BDD5 20EF    JR NZ,FCHEC1
BDD7 D1      POP DE
BDD8 C1      POP BC
BDD9 A7      AND A          ;CY=0
BDDA C9      RET

;
BDD8 D1      FCHEC3: POP DE
BDDC C1      POP BC
BDDD D1      POP DE
BDDE C1      POP BC
BDDF 37      SCF            ;CY=1 ショウカ"イフ"ツ アリ
BDE0 C9      RET

;
; カソク フィールド"ノ DATA ラ モトネル IN E=X,D=Y FILED DATA D000H-
;                      OUT A
;
BDE1 D5      GETWAD: PUSH DE
BDE2 E5      PUSH HL
BDE3 62      LD H,D
BDE4 2E50    LD L,80
BDE6 C00BE   CALL 0BE00H    ;Y*80
BDE9 1600    LD D,0
BDEB 19      ADD HL,DE      ;+X
BDEC 1100D0  LD DE,0D000H   ;DATA TOP
BDEF 19      ADD HL,DE
BDF0 7E      LD A,(HL)
BDF1 E1      POP HL
BDF2 D1      POP DE
BDF3 C9      RET
;
;

```

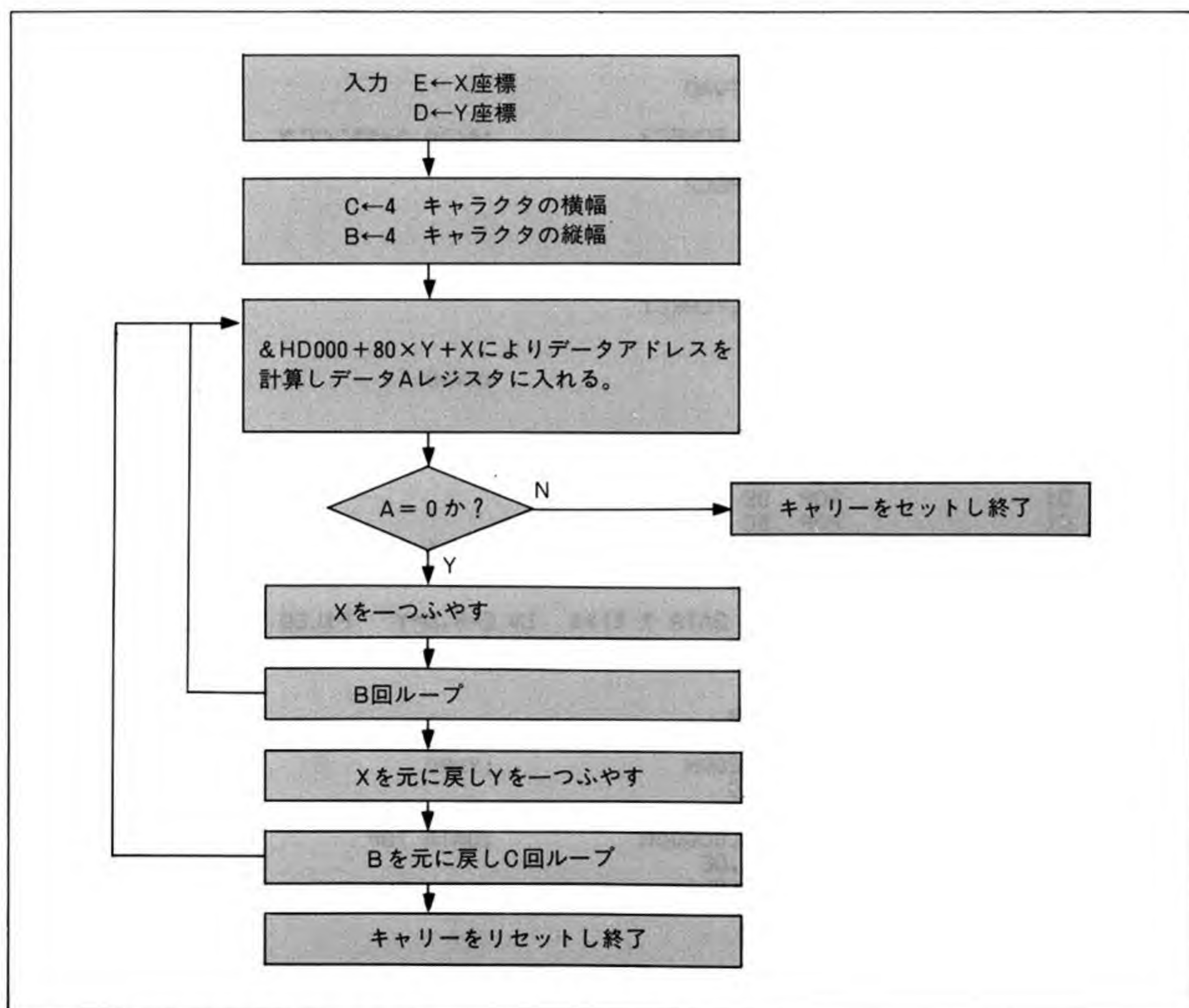


リスト6-7の&HBD00からが、自分のキャラクタが動いた時に、他のキャラクタに当たったかどうか調べるものです。もし、他と衝突したならば、お互いの方向を見て、向い合う状態ならば自分の負けとし、それ以外は自分の勝ちとします。第6-4図に作業手順がありますので参考にして下さい。

リスト6-7の&HBD70からは、敵が動いた時に、プレイヤーのキャラクタと衝突したかどうかを見るもので、もし衝突しているならば敵の勝ちとして、&HE000番地に"0"を代入します。

上の二つのプログラムは、&HBD7Bからのサブルーチンを共通に使っています。このルーチンはHLレジスタに、敵キャラクタのワークエリア先頭アドレスを代入してサブルーチンコールすると、衝突しているならキャリーフラグを1に、していないなら0にして戻って来ます。

第6-5図の様に、XY座標それぞれの差が±3以内ならば衝突したと言う事になり、それを調べるのに、差が正になる時と負になる場合をキャリーフラグで調べて場合わけし、正になる時は4より少なければ衝突、負になる時は-3、すなわち&HFD以上ならば衝突したと判定しています。



第6-3図 リスト6-6「仮想空間とXYとの判定」手順



## リスト6-7 衝突判定

```

;
; アタック ハンティ
;
; OUT FLAG
;
; ORG 0BD00H
;
BD00 2110E0 AHANT: LD HL,0E010H
BD03 0603 LD B,3
BD05 C5 AHANT1: PUSH BC
BD06 E5 PUSH HL
BD07 CD7BBD CALL HANTS
BD0A E1 POP HL
BD0B C1 POP BC
BD0C 3807 JR C,AHANT2
BD0E 111000 LD DE,10H
BD11 19 ADD HL,DE ;NEXT MONSTER
BD12 10F1 DJNZ AHANT1
BD14 C9 RET

;
BD15 23 AHANT2: INC HL
BD16 23 INC HL
BD17 23 INC HL ;HL+3
BD18 3A03E0 LD A,(0E003H) ;READ DIR.
BD1B 3D DEC A
BD1C 2007 JR NZ,AHANT3
BD1E 7E LD A,(HL) ;IF DIR=1
BD1F FE03 CP 3
BD21 281C JR Z,EWIN
BD23 181F JR MWIN

;
BD25 3D AHANT3: DEC A
BD26 2006 JR NZ,AHANT4
BD28 7E LD A,(HL) ;IF DIR=2
BD29 A7 AND A
BD2A 2813 JR Z,EWIN
BD2C 1816 JR MWIN

;
BD2E 3D AHANT4: DEC A
BD2F 2007 JR NZ,AHANT5
BD31 7E LD A,(HL) ;IF DIR=3
BD32 FE01 CP 1
BD34 2809 JR Z,EWIN
BD36 180C JR MWIN

;
BD38 7E AHANT5: LD A,(HL) ;IF DIR=0
BD39 FE02 CP 2
BD3B 2802 JR Z,EWIN
BD3D 1805 JR MWIN

;
BD3F AF EWIN: XOR A
BD40 3200E0 LD (0E000H),A ;テキノカチ
BD43 C9 RET

;
BD44 2B MWIN: DEC HL
BD45 2B DEC HL
BD46 2B DEC HL ;HL=WORK TOP
BD47 3600 LD (HL),0 ;RESET
BD49 CD50BF CALL 0BF50H ;CHR CLS

```



```

BD4C 3A09E0      LD    A,(0E009H)      ;READ POINT
BD4F 3C          INC    A
BD50 3209E0      LD    (0E009H),A      ;SET POINT
BD53 C9          RET

;
; テキ アタック ハンティ          IN HL WORK TOP
;                                OUT FLAG
;
;                                ORG 0BD70H
;
BD70 E5          EHANT: PUSH HL
BD71 CD7BBD      CALL HANTS
BD74 E1          POP  HL
BD75 D0          RET  NC

;
BD76 AF          XOR   A
BD77 3200E0      LD    (0E000H),A      ;LIFE FLAG=0
BD7A C9          RET

;
BD7B 7E          HANTS: LD    A,(HL)      ;OUT CY
BD7C A7          AND   A
BD7D C8          RET  Z                ;FLAG CHECK CY=0

;
BD7E 23          INC   HL
BD7F 5E          LD    E,(HL)          ;READ XT
BD80 23          INC   HL
BD81 56          LD    D,(HL)          ;READ YT
BD82 2101E0      LD    HL,0E001H      ;MY WORK X
BD85 4E          LD    C,(HL)          ;READ XM
BD86 23          INC   HL
BD87 46          LD    B,(HL)          ;READ YM

;
BD88 79          LD    A,C            ;CHECK X
BD89 93          SUB   E              ;XD=XM-XT
BD8A 3806        JR    C,EHANT1
BD8C FE04        CP    4
BD8E 3016        JR    NC,EHANTP      ;IF XD>=4 THEN PASS
BD90 1804        JR    EHANTA        ;0<=XD<=3

;
BD92 FEFD        EHANT1:CP    -3
BD94 3810        JR    C,EHANTP      ;IF XD>-3 THEN PASS

;
BD96 78          EHANTA:LD    A,B      ;CHECK Y
BD97 92          SUB   D              ;YD=YM-YT
BD98 3806        JR    C,EHANT2
BD9A FE04        CP    4
BD9C 3008        JR    NC,EHANTP      ;IF YD>=4 THEN PASS
BD9E 1804        JR    EHANTB        ;0<=YD<=3

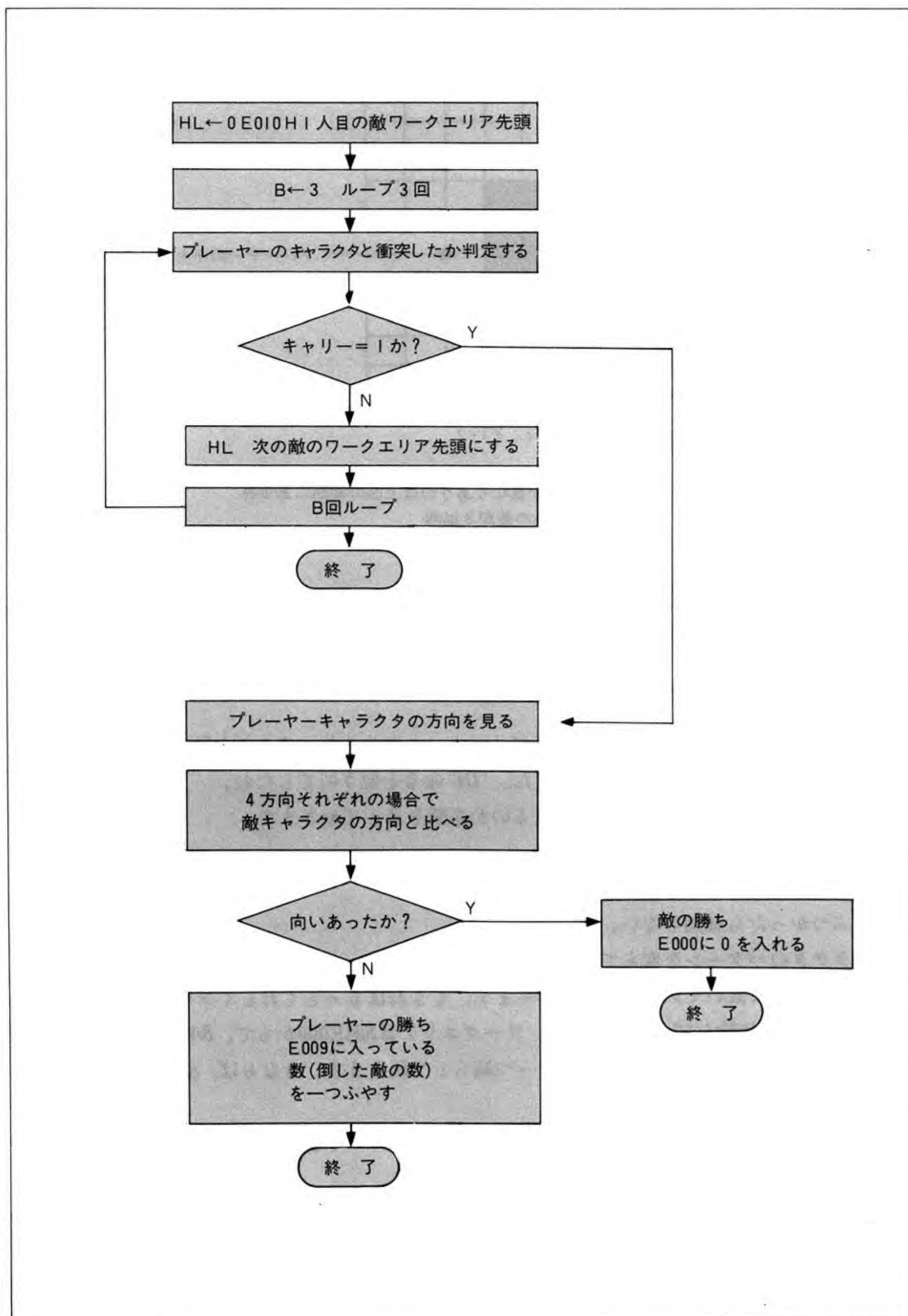
;
BDA0 FEFD        EHANT2:CP    -3
BDA2 3802        JR    C,EHANTP      ;IF YD>-3 THEN PASS

;
BDA4 37          EHANTB:SCF
BDA5 C9          RET                ;CY=1 ATTACK

;
BDA6 A7          EHANTP:AND  A
BDA7 C9          RET                ;CY=0 PASS

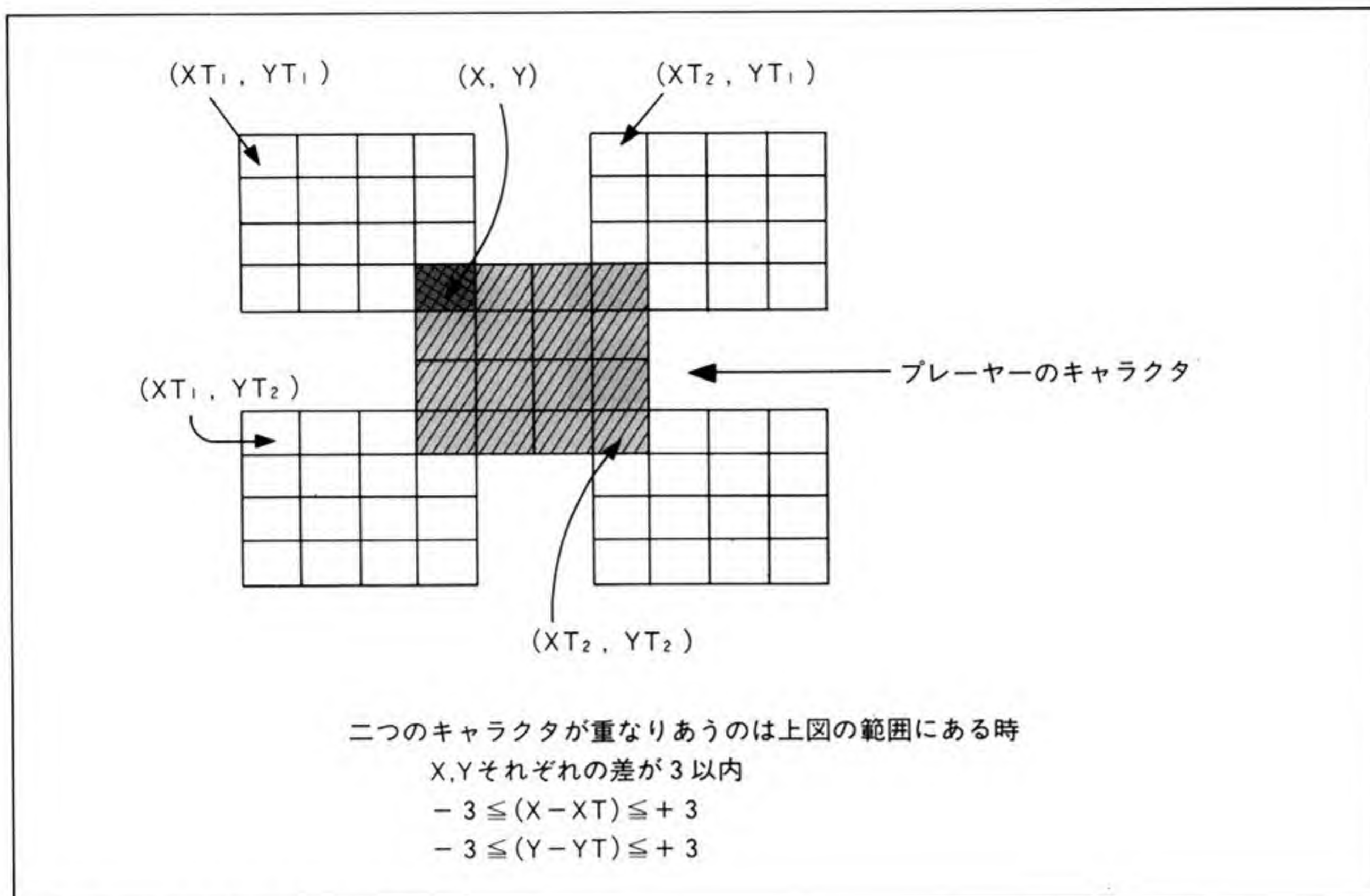
```





第6-4図 リスト6-7「衝突判定」手順





第6-5図 キャラクタ同志の衝突

#### (Ⅳ) キー入力で自分のキャラクタを動かす

最低必要なサブルーチンがそろった所で、いよいよ自分のキャラクタを動かしてみましょう。

キー入力の方法は前に説明いたしました。“IN”命令を使うのでしたね。

プログラムを作る前に、まず何をさせるのか手順を考えておきます。

- 移動方向は簡単に、四方向のみとする。
- 画面からはみ出さないようにする。
- 壁にぶつかったら移動しない。
- キャラクタのパターンを変えて書き直す。

以上の事を頭に置いてプログラムを作ります。できればもっとくわしくフローを書きましょう。

リスト6-8が具体的なプログラムです。ワークエリアは&HE000からで、&HE007の数字が“0”の時にのみ移動させ、0以外ならば数を一つ減らして戻ります。0ならば、&HE007に&HE008の値を代入します。

つまり&HE008に入れてある数だけ何もしないでリターンしますが、これは他の移動キャラクタとの速さの比を様々に変えられるようにしたためです。

移動の仕組は単純ですから、リストおよび第6-6図の作業手順を見ていただければ、お分かりいただけると思います。これに斜め移動も加えればもっと動きやすくなるでしょう。

気を付けていただきたいのは、XとYの範囲が、0～79と0～49ではなく、0～76と0～46にな







```

;
BC44 7B      MYMR: LD  A,E          ;READ X
BC45 FE4C    CP   76          ;RIGHT LIMMIT 76
BC47 C8      RET  Z
;
BC48 1C      INC  E           ;X=X+1
BC49 3E01    LD   A,1
BC4B 3203E0  LD   (0E003H),A   ;SET D=1
BC4E 1809    JR   MYMOV3      ;GOTO DRAW
;
BC50 7B      MYML: LD  A,E          ;READ X
BC51 A7      AND  A           ;LEFT LIMMIT 0
BC52 C8      RET  Z
;
BC53 1D      DEC  E           ;X=X-1
BC54 3E03    LD   A,3
BC56 3203E0  LD   (0E003H),A   ;SET D=3
;
BC59 CDC0BD  MYMOV3:CALL 0BDC0H   ;FILED CHECK
BC5C 3813    JR   C,MYMOV5
;
BC5E 3A04E0  LD   A,(0E004H)    ;POSE 0,1 CHANGE
BC61 3D      DEC  A
BC62 2802    JR   Z,MYMOV4
BC64 3E01    LD   A,1
BC66 3204E0  MYMOV4:LD   (0E004H),A
;
BC69 7B      LD   A,E
BC6A 3201E0  LD   (0E001H),A    ;SET X
BC6D 7A      LD   A,D
BC6E 3202E0  LD   (0E002H),A    ;SET Y
BC71 2100E0  MYMOV5:LD   HL,0E000H
BC74 E5      PUSH HL
BC75 CD50BF  CALL 0BF50H        ;CHR CLS
BC78 E1      POP  HL
BC79 CD00BF  CALL 0BF00H        ;CHR PUT
BC7C CD00BD  CALL 0BD00H        ;HANTEI
BC7F C9      RET

```

っていることです。これはキャラクタにX,Y四つずつの大きさがあるためです。

また、&HBC75と&HBC79のクリアルーチンとプットルーチンのへだたりを見て下さい。

ゲームを作り始めた頃は、

古いキャラクタを消す→座標を移動する→判定する→新しいキャラクタを書く。

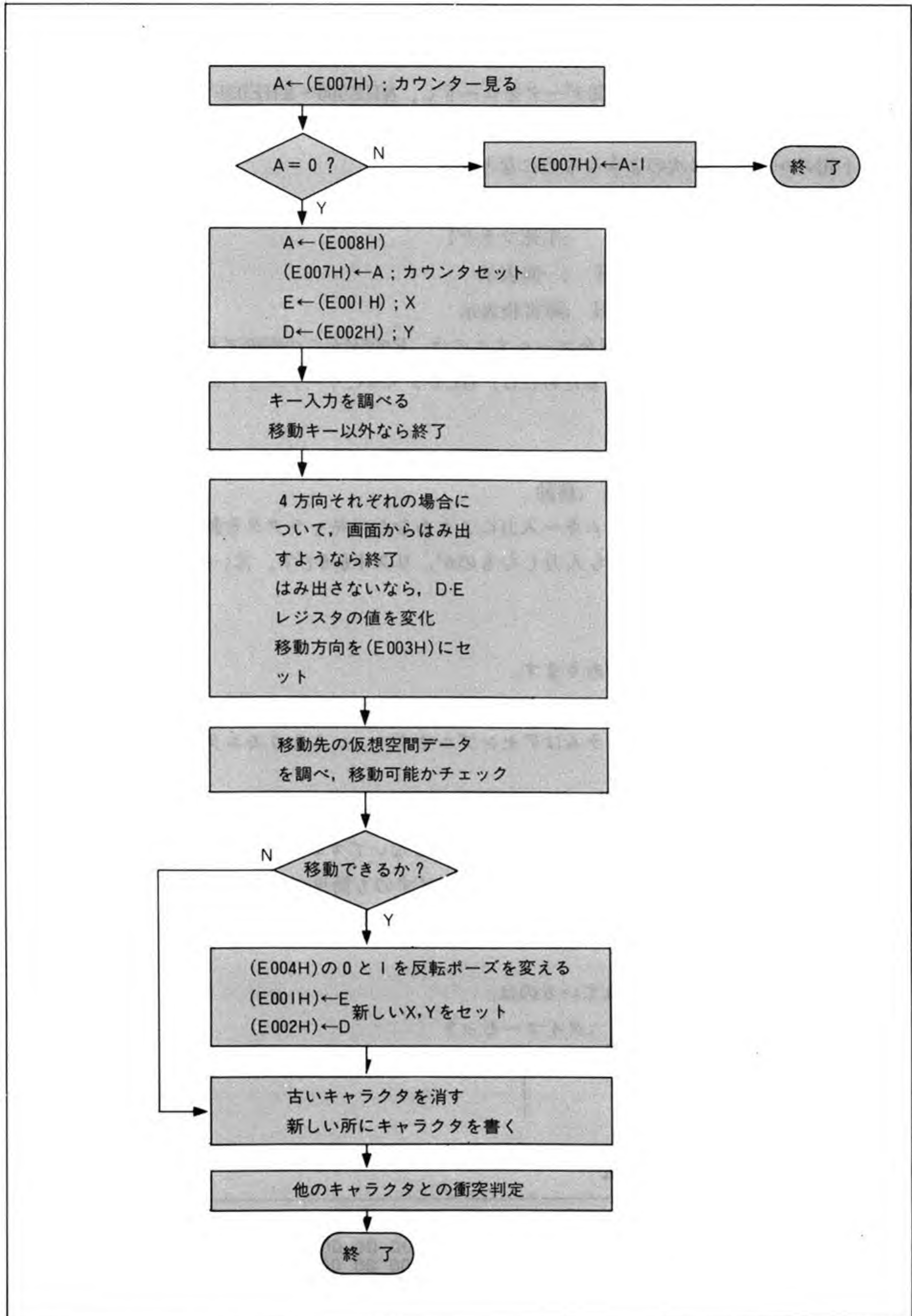
と言う手順で、消してから書くまでの間に他の処理が入っていたのですが、チラつきを防ぐため、この間はできるだけ短い方が良いでしょう(もちろん、やむをえず長くなる時もある)。

そこで、&HE005、&HE006に書いた画面座標をストックしておく意味が、単に計算の手間をばくだけでない事がわかりいただけるでしょう。

では、このサブルーチンをテストプレイしてみましょう。使うプログラムはリスト6-1からリスト6-8までのマシン語プログラムです。


まずキャラクタデータを&H6000番地からロードしますが、これはディスクにセーブしてあるデ





第6-6図 リスト6-8「キー入力による移動」手順



ータを、BLOAD"ファイル名", &H6000 

とすればOKです。

さらに、&HD000からの仮想空間データをロードし、&HE000~&HE03Fまでのワークエリアを全部0にしておいて下さい。

テスト用のルーチンは次のようなものになります。

21 00E0 LD HL, 0E000H ;ワークエリア先頭

36 01 LD (HL), 1 ;生死フラグ1

CD 00BF CALL 0BF00H ;一回表示

CD A0BC CALL 0BCA0H ;障害物表示

ここまでが下準備です。BF00Hをコールするのは、E005Hからの画面アドレスをセットするためです。このルーチンをコールするためには、HLレジスタにワークエリアの先頭番地を入れておく必要がありました。

そして、

CD 00BC CALL 0BC00H ;移動

をくり返し行えば、リアルタイムキー入力によりあなたのキャラクタを動かせます。

以上のプログラムを&HB800から入力したものが、リスト6-9です。言い遅れましたが、この実験には、

CLEAR, &HB7FF 

をあらかじめ行っておく必要があります。

リスト6-9を見ると今までのプログラムと違ってプログラム中に"00"が多く入っているのに気づかれるでしょう。実はこのプログラムはアセンブルせずに、いきなりモニタのEコマンドで書き込んだものです。

そこでわかりやすい番地にするために、"00"が多くなったのです。"00"はニーモニックで"NOP", つまりノーオペレーションの事で、CPUは何もしないですから、プログラムとプログラムの間を"00"で埋めておけば、後からプログラムをつけ足すのも簡単です。

このプログラムの"余白"とも言える部分に、"00"以外の数があると、バグの原因になる事もありますので注意して下さい。

リスト6-9の&HB820から書かれているのは、

01 0010 LD BC, 10000H ;タイマーセット

0B DEC BC

78 LD A, B



リスト6-9 実行テスト用プログラム

B800	21	00	E0	36	01	CD	00	BF	CD	A0	BC	00	00	00	00	00	!	=	6	^	ソ	ハ	シ
B810	CD	00	BC	00	00	00	00	00	00	00	00	00	00	00	00	00	ハ	シ					
B820	01	00	10	0B	78	B1	C2	23	B8	00	00	00	00	00	00	00							
B830	C3	10	B8	00	00	00	00	00	00	00	00	00	00	00	00	00							

xアツ#ク  
テ ク



```

B1          OR  C
C2 23B8  JP  NZ, 0B823H

```

と言うプログラムで、BCに1000Hを入れてこの数だけ空転させ、時間待ちをしています。

これは移動が早すぎてしまうのをふせぐためのタイマーです。好みに合わせてBC レジスタにセットする値を変えて下さい。0001で最も早く、0000で最も遅くなるでしょう。また早いと、キャラクタのチラツキも起きてきます。

&HB830は、"JP 0B810H"で、無条件に移動ルーチンをコールしている所へ戻り、無限にループします。ジャンプ命令に相対ジャンプを使わないのは、絶対ジャンプの方がモニタで見分かりやすいからです。もちろん相対ジャンプに直してもかまいません。

では、クリアキーでテキスト画面をクリアしてから、

```
h]GB800
```

とやってみて下さい。無限ループを作っていますから、テストを中断させるにはリセットスイッチを押します。

おかしくなってしまった時は、もう一度、ジャンプ先やコール先のアドレスをチェックし、もしキャラクタを移動させて画面からはみ出してしまうような時には、リセット後、&HE000からのワークエリアの内容を見て、それぞれの数値が設定した範囲(たとえば、Xなら0から76まで)になっているかをチェックします。

一つ一つのサブルーチンが確実に動作するならば、大きなバグは出ないと思います。ブロックを積み重ねるように一つずつ作って行けば、どんな大きなプログラムでも作り上げられるのです。

## (V) データで移動させる

次にキャラクタをデータで移動させる方法を考えてみましょう。

その方法はいくつかあります。

- 移動先のxy座標(あるいはGRAMの番地)をデータにして並べ、それを見て行く。画面のどこでもジャンプできるが、移動のし方がワンパターンになる。
- 移動する大きさを、xyについてデータにする。

元いた位置のx, yに移動データを足し算して、次のx, yを決めるので、始めの位置が違えば同じデータでも移動場所が異なる。データの数値は符号つき(+, -)の数値になる。"ベクトル"を学習されている方には、"ベクトルのxy成分をデータにする"と考えていただければわかると思います。

私の場合は以上のようなデータ設定が多いのですが、キャラクタの"向き"の判定が面倒なので、今回は"一度に、四方向どれか、一つだけ動く"ものとして、

- 移動方向のみをデータにする。

と言う方法を使ってみましょう。きわめて単純ですが、データの作り方によって様々な動き方をさせられます。データの種類は四方向ですから四つの数字で足ります。始めに設定したキャラクタの方向に対応させて、

```

0:下移動,   1:右移動
2:上移動,   3:左移動   としましょう。

```



次にデータをどこに格納するか、メモリの割り付けをします。&HC000～&HCFFFが空いているので、そこに置きましょう。

たとえば次の様に設定してみました。

&HC000～&HC3FF：上から出て主に下方向へ行く

&HC400～&HC7FF：下から出て主に上方向へ行く

&HC800～&HCBFF：左から出て主に右方向へ行く

&HCC00～&HCFFF：右から出て主に左方向へ行く

キャラクタが上下左右のどこからか出現すると考えると、出てすぐに引っ込んでしまうのではなさけないので、上の四つに場合分けしました。一つの所から出現するのであれば、また違う設定になるでしょう。

さて、出現してから消えるまでいくつのデータが必要になるでしょうか？ それはデータの内容により大分異なると思いますが、真っすぐ画面を横切るのに76歩ですから、ここで設定したように1Kバイトのデータは必要ないでしょう。


そこで、各々の場合について、さらに四通りにデータを分けてみます。

たとえば、&HC000～&HC3FFのデータを、&HC000～&HC0FF、&HC100～&HC1FF、&HC200～&HC2FF、&HC300～&HC3FFの四つに分けてしまえば、上から出現したキャラクタの移動のパターンを、四通り持つ事ができるようになります。

もっと細かくもできると思いますが、この程度にしておきました。

さて、移動データを作るのは、仮想空間の設定をした時の様に、簡単なBASICによるツールを作ればよいのですが、それは皆さんで考えて下さい。

リスト6-10に四つの場合のサンプルデータをのせておきます。これはモニタで直接打ち込んだものです。たとえば、下から出現する時のデータは、

h]FC400, C7FF, 2 

とやって、上への移動データ"2"をデータエリア全部に入れておき、Eコマンドで適当に他の

リスト6-10 サンプル移動データ

#### 下移動用

```

C000 00 00 00 00 00 00 01 01 01 00 00 00 03 03 03 03 03
C010 03 02 02 02 01 01 01 01 00 00 00 00 00 00 00 00 00
C020 00 00 00 00 00 00 00 00 00 00 00 00 03 03 03 03 00
C030 00 01 01 01 01 01 01 01 01 01 01 01 01 01 03 03 03
C040 00 00 00 00 00 00 00 00 00 00 00 00 00 03 03 03 03
C050 03 03 03 03 03 03 03 03 03 03 03 00 00 00 00 00 00
C060 00 00 00 00 00 00 02 02 02 02 02 02 02 01 01 01 01
C070 01 01 01 01 01 01 00 00 00 00 00 00 00 00 00 00 00
C080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```



## 上移動用

```

C400 02 02 02 02 02 02 01 01 01 01 01 02 02 03 03 03
C410 02 01 01 01 01 01 02 02 03 03 03 03 03 03 03 03
C420 02 02 02 02 02 02 01 01 01 00 00 00 00 00 01 01
C430 01 01 02 02 02 02 02 02 02 02 02 02 02 02 02 03
C440 03 03 03 03 03 03 03 03 03 02 02 02 02 02 02 02
C450 02 02 02 02 02 02 01 01 01 01 01 02 02 02 02 02
C460 03 03 03 03 03 03 03 03 03 02 02 02 02 02 02 02
C470 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C480 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C490 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4A0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4B0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4C0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4D0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4E0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
C4F0 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02

```

## 右移動用

```

C800 01 01 01 01 01 01 01 01 03 03 03 02 02 02 02 01
C810 01 01 01 01 01 00 00 00 02 02 01 01 01 01 01 01
C820 01 01 01 02 02 02 02 02 01 01 01 01 02 02 02 02
C830 01 01 01 01 01 02 02 02 02 02 01 01 01 01 01 01
C840 01 00 00 00 00 01 01 01 01 01 01 00 00 00 00 00
C850 00 00 00 00 03 03 03 03 03 03 02 02 02 02 01 01
C860 01 01 01 01 00 00 00 00 00 01 01 01 01 01 01 01
C870 01 01 02 02 02 02 02 02 02 00 00 00 00 01 01 01
C880 01 01 01 01 02 02 02 02 02 02 02 01 01 01 01 01
C890 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8A0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8B0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8C0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8D0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8E0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
C8F0 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01

```

## 左移動用

```

CC00 03 03 03 03 03 03 00 00 00 00 03 03 03 03 02 02
CC10 02 02 02 03 03 03 03 03 00 00 00 00 03 03 03 03
CC20 03 03 03 03 01 01 01 01 03 03 03 03 03 02 02 02
CC30 02 02 02 02 02 02 02 03 03 03 03 03 00 00 00 00
CC40 00 00 00 00 00 00 00 00 00 00 03 03 03 03 03 03
CC50 03 03 03 03 03 02 02 02 02 02 02 02 02 02 01 01
CC60 01 01 01 03 03 03 03 03 03 03 03 03 03 03 03 03
CC70 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CC80 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CC90 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCA0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCB0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCC0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCD0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCE0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03
CCF0 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03 03

```



方向の数字を書き込みました。

作ったデータは忘れずに、ディスクにセーブしておきましょう。

データができた所で、移動の方法を考えてみましょう。大まかに考えて次のようになります。

●移動データを見る。

●その方向へ動かす。もし、画面からはみ出すようなら終りにする。

とても単純ですが、移動データを見るには、どこの(何番地の)データかわかるようにしてやらねばなりません。その番地を示す数字もワークエリアに作っておきましょう。

ワークエリアのアドレスを、&HE010からにすると、大部分は“キー入力”の所でやったのと同様でよいのですが、E019番地から2バイトにデータアドレスを入れる事にします(第6-2表参照)。

それにもう一つ大事な事を忘れていた様ですね。“画面からはみ出したら終り”はよいのですが、“始め”がありません。どこからスタートするのか決めなければ移動のさせようがありませんね。

キャラクタを移動させるには、まずスタート位置等の決定が必要です。つまり変数などをイニシャライズ(初期値化)するのです。

この事は“キー入力移動”の所では出て来ないようでしたが、実はちゃんとやっているのです。

CALL 0BF00H などの下準備と“ワークエリアを”00”で埋める”，と言うのがありました。つまり、X=0, Y=0からスタートさせていたわけです。

移動ルーチンを作るのに先立って、まずスタート位置を決定するプログラムを作りましょう。リスト6-11がそのプログラムで、ワークエリアの先頭アドレスをHLレジスタに入れてコールすると、まずランダムで0~3の方向を示す数字を作ります。

0の時は下向きだから、画面上よりスタート : Y = 0

1の時は右向きだから、画面左よりスタート : X = 0

2の時は上向きだから、画面下よりスタート : Y = 46

3の時は左向きだから、画面右よりスタート : X = 76

これら四つの辺のどのあたりから出現するかは、またランダムで決定します。

$18 \leq X \leq 57, \quad 8 \leq Y \leq 37$

の間になるように設定しています。


では、いよいよ、データ移動のプログラミングに移りましょう。もし、ワークエリア先頭の&HE010の値が”0”なら、イニシャライズルーチンに飛ばすようにしました。リスト6-12がプログラムで、作業の手順は第6-7図にあります。

動作の確認をする時には、(Ⅳ)のキー入力移動のテストの様に、必要なデータとプログラムをロードし、

CLEAR, &HB7FF 

を行って、リスト6-9の実行テスト用のプログラムを打ち込みます。ただし、&HB812番地の”BC”は、”BB”に直して、CALL 0BB00Hにします。

そこでモニタから、

h]GB810 

とやると、キャラクタが移動を始め、スクリーンのはじへ行くと消えて、また別の場所から出



リスト6-11 スタート位置決定プログラム

```

;
; SET START X,Y IN HL,WORK TOP
;
;
;          ORG  0BBC0H
;
BBC0 0E04      SETST: LD   C,4
BBC2 CD20BE    CALL  0BE20H          ;RAND(4)
BBC5 51        LD    D,C            ;D=C
BBC6 79        LD    A,C
BBC7 3D        DEC   A
BBC8 281C      JR    Z,SETSTR        ;TO RIGHT
BBCA 3D        DEC   A
BBCB 2807      JR    Z,SETSTU        ;TO UP
BBCD 3D        DEC   A
BBCE 281A      JR    Z,SETSTL        ;TO LEFT
;
BBD0 1E00      LD    E,0            ;TO DOWN
BBD2 1802      JR    SETSUD          ;Y=0
;
BBD4 1E2E      SETSTU:LD   E,46      ;Y=46
;
BBD6 0E28      SETSUD:LD   C,40
BBD8 CD20BE    CALL  0BE20H
BBD8 3E12      LD    A,18
BBD0 81        ADD   A,C            ;START X=18+RAND(40)
BBDE 3601      LD    (HL),1         ;ON
BBE0 23        INC   HL
BBE1 77        LD    (HL),A         ;X=A
BBE2 23        INC   HL
BBE3 73        LD    (HL),E         ;Y=E
BBE4 1814      JR    SETSTE
;
BBE6 1E00      SETSTR:LD   E,0
BBE8 1802      JR    SETSRL          ;X=0
;
BBEA 1E4C      SETSTL:LD   E,76     ;X=76
;
BBEC 0E1E      SETSRL:LD   C,30
BBEE CD20BE    CALL  0BE20H
BBF1 3E08      LD    A,8
BBF3 81        ADD   A,C            ;START Y=8+RAND(30)
BBF4 3601      LD    (HL),1         ;ON
BBF6 23        INC   HL
BBF7 73        LD    (HL),E         ;X=E
BBF8 23        INC   HL
BBF9 77        LD    (HL),A         ;Y=A
;
BBFA 23        SETSTE:INC  HL
BBFB 72        LD    (HL),D         ;SET DIR
BBFC C9        RET
;
;

```



リスト6-12 データによる移動

```

;
; MOVE CHR BY DATA      XD=(E011H)      DATA DOWN C000H
;                          YD=(E012H)      UP C400H
;                          DD=(E013H)      RIGHT C800H
;                          PD=(E014H)      LEFT CC00H
;                          WD1=(E017H)     DATA ADDRESS IN E019H
;                          WD2=(E018H)
;
; ORG 0BB00H
;
BB00 3A10E0  DAMOV: LD  A,(0E010H)      ;READ SET FLAG
BB03 A7      AND  A
BB04 2003    JR   NZ,DAMOV0
BB06 C37FBB  JP   INIDAM              ;INITIALIZE MOVE CHR.
;
BB09 3A17E0  DAMOV0:LD  A,(0E017H)      ;READ WAIT 1
BB0C A7      AND  A
BB0D 2805    JR   Z,DAMOV1
BB0F 3D      DEC  A                  ;WAIT COUNT-1
BB10 3217E0  LD   (0E017H),A
BB13 C9      RET
;
BB14 3A18E0  DAMOV1:LD  A,(0E018H)      ;READ WAIT 2
BB17 3217E0  LD   (0E017H),A          ;SET WAIT COUNT
;
BB1A 3A11E0  LD   A,(0E011H)          ;READ XD
BB1D 5F      LD   E,A                ;E=X
BB1E 3A12E0  LD   A,(0E012H)          ;READ YD
BB21 57      LD   D,A                ;D=Y
;
BB22 2A19E0  LD   HL,(0E019H)          ;READ DATA ADDRESS
BB25 7E      LD   A,(HL)              ;READ DIRECT.
BB26 23      INC  HL
BB27 2219E0  LD   (0E019H),HL          ;SET NEXT ADDRESS
BB2A 3213E0  LD   (0E013H),A          ;SET D
BB2D 3D      DEC  A
BB2E 2815    JR   Z,DAMR              ;GOTO RIGHT
BB30 3D      DEC  A
BB31 280B    JR   Z,DAMU              ;GOTO UP
BB33 3D      DEC  A
BB34 2817    JR   Z,DAML              ;GOTO LEFT
;
BB36 7A      LD   A,D                ;GOTO DOWN
BB37 FE2E    CP   46                  ;DOWN LIMMIT 46
BB39 283B    JR   Z,DAMOFF
BB3B 14      INC  D                  ;Y=Y+1
BB3C 1814    JR   DAMOV3              ;GOTO DRAW
;
BB3E 7A      DAMU: LD  A,D            ;READ Y
BB3F A7      AND  A
BB40 2834    JR   Z,DAMOFF
BB42 15      DEC  D                  ;Y=Y-1
BB43 180D    JR   DAMOV3              ;GOTO DRAW
;
BB45 7B      DAMR: LD  A,E            ;READ X
BB46 FE4C    CP   76                  ;RIGHT LIMMIT 76
BB48 282C    JR   Z,DAMOFF
BB4A 1C      INC  E                  ;X=X+1
BB4B 1805    JR   DAMOV3              ;GOTO DRAW
;
BB4D 7B      DAML: LD  A,E            ;READ X
BB4E A7      AND  A                  ;LEFT LIMMIT 0
BB4F 2825    JR   Z,DAMOFF
BB51 1D      DEC  E                  ;X=X-1
;
BB52 3A14E0  DAMOV3:LD  A,(0E014H)      ;POSE 0,1 CHANGE

```



```

BB55 3D          DEC  A
BB56 2802        JR   Z,DAMOV5
BB58 3E01        LD   A,1
BB5A 3214E0      DAMOV5:LD (0E014H),A
;
BB5D 7B          LD   A,E
BB5E 3211E0      LD   (0E011H),A      ;SET X
BB61 7A          LD   A,D
BB62 3212E0      LD   (0E012H),A      ;SET Y
BB65 2110E0      LD   HL,0E010H
BB68 E5          PUSH HL
BB69 E5          PUSH HL
BB6A CD50BF      CALL 0BF50H          ;CHR CLS
BB6D E1          POP  HL
BB6E CD00BF      CALL 0BF00H          ;CHR PUT
BB71 E1          POP  HL
BB72 CD70BD      CALL 0BD70H          ;HANTEI
BB75 C9          RET
;
BB76 2110E0      DAMOFF:LD HL,0E010H
BB79 3600        LD   (HL),0          ;OFF
BB7B CD50BF      CALL 0BF50H          ;CHR CLS
BB7E C9          RET
;
; INITIALZE DATA MOVE CHR.
;
BB7F 2110E0      INIDAM:LD HL,0E010H
BB82 CDC0BB      CALL SETST          ;SET START POINT
BB85 3A13E0      LD   A,(0E013H)      ;READ DIR.
BB88 3D          DEC  A
BB89 2005        JR   NZ,INIDA1
BB8B 1100C8      LD   DE,0C800H      ;RIGHT DATA TOP
BB8E 1813        JR   INIDA4
;
BB90 3D          INIDA1:DEC A
BB91 2005        JR   NZ,INIDA2
BB93 1100C4      LD   DE,0C400H      ;UP DATA TOP
BB96 180B        JR   INIDA4
;
BB98 3D          INIDA2:DEC A
BB99 2005        JR   NZ,INIDA3
BB9B 1100CC      LD   DE,0CC00H      ;LEFT DATA TOP
BB9E 1803        JR   INIDA4
;
BBA0 1100C0      INIDA3:LD DE,0C000H  ;DOWN DATA TOP
;
BBA3 0E04        INIDA4:LD C,4
BBA5 CD20BE      CALL 0BE20H          ;RAND(4)
BBA8 7A          LD   A,D
BBA9 81          ADD  A,C            ;ADDRESS HIGH+RAND(4)
BBAA 57          LD   D,A
BBAB ED5319E0    LD   (0E019H),DE    ;DATA ADR. SET
BBAF 2110E0      LD   HL,0E010H
BBB2 CD00BF      CALL 0BF00H          ;DRAW
BBB5 C9          RET

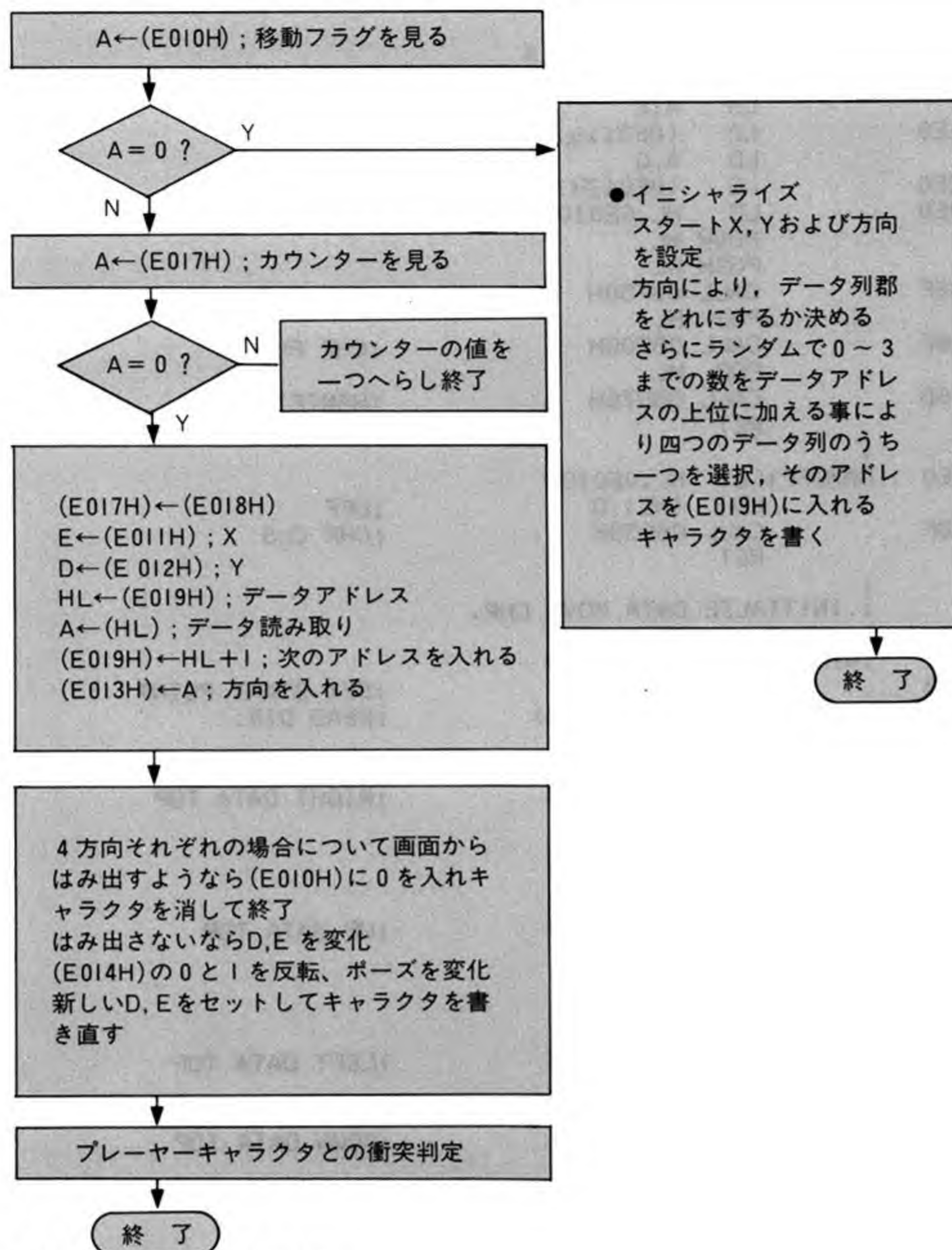
```

現すると思います。もし、動きがワンパターンだったりするようでしたら、リセットしてプログラムを見直します。まさか、移動データをロードし忘れたなんて事はありませんよね。

#### h)GB800

とすれば、前のキャラクタと障害物が表示されますが、データ移動のキャラクタは障害物に当たってもかまわずに進んでしまいます。そう、第6-7図を見てもおわかりかと思いますが、このプログ





ラムには、障害物を判定する部分がないのです。プログラムが複雑になるので避けましたが、やろうと思えばできない事はありません。ぶつかった所で方向を変えて他のデータ列に切り変えるなどの方法があります。やってみてください。

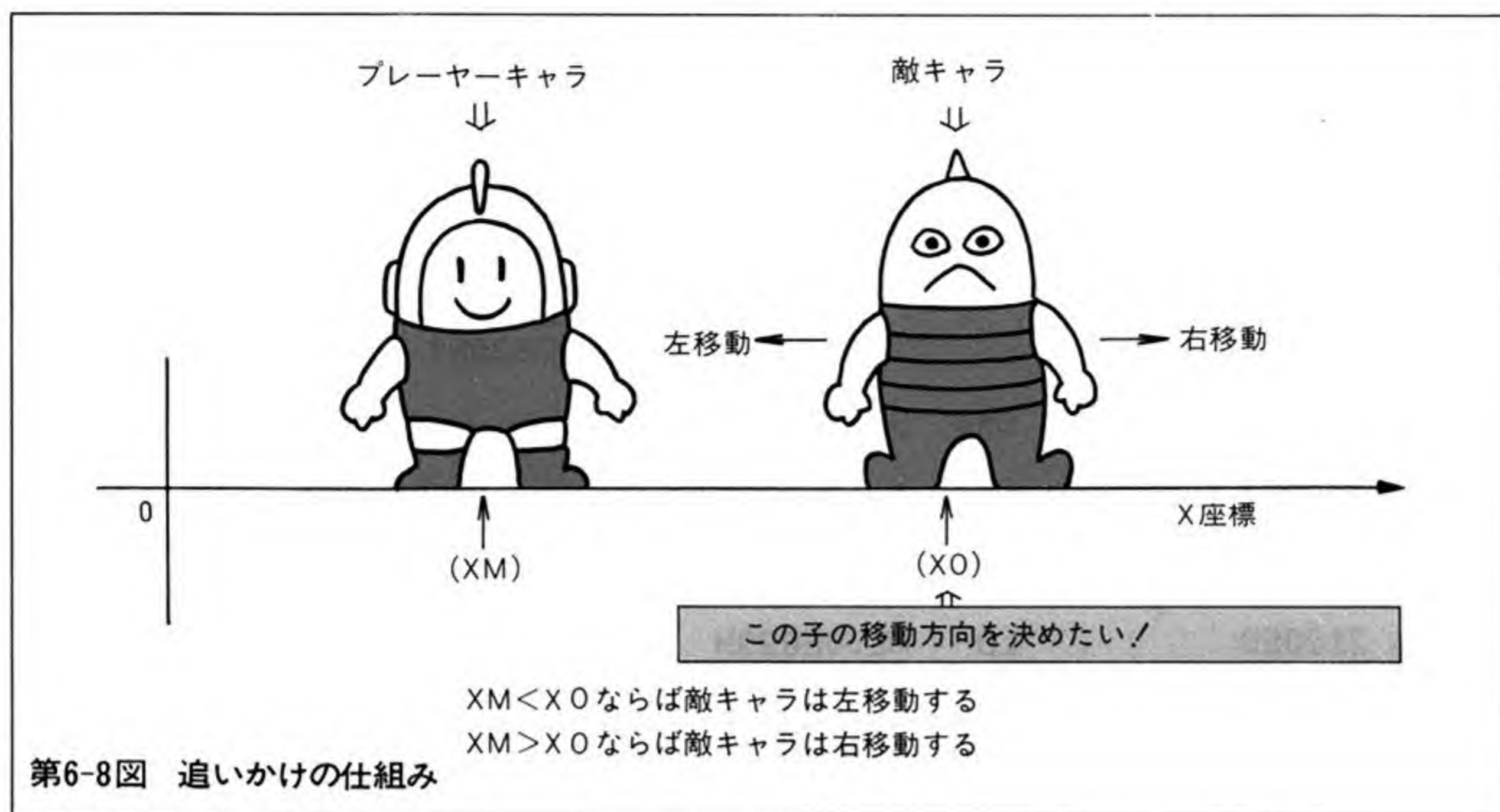
## V) 追いかけて移動する

ゲームにはある程度のスリルが必要です。スピードもそうですが、自分のキャラクタをねらって追いかけて来る、なんてものもぜひほしいですね。

そこで、追いかける仕組みを考えてみましょう。

今、簡単に考えて、一直線にプレイヤーのキャラクタと、敵のキャラクタが並んでいるものとします(第6-8図)。





これから考えるのは、敵キャラクタをどちら向きに動かすかです。敵キャラクタを中心に考えて下さい。

もし、プレイヤーのキャラクタが左にいて、敵のキャラクタが右の時、敵キャラクタを左向きに移動させれば、一歩追いかけた事になりますね。

どちらが右で、どちらが左かを知るには、X座標をくらべればわかります。プレイヤーの座標をXM、敵の座標をXOとすると、 $XM < XO$  ならば、プレイヤーのキャラクタが左側である事がわかります。

Y座標についても同じ事を考えます。こうして、XY別々にどの方向へ動くか決める事ができますが、さて、この後の処理には何通りかのやり方があります。

- XYともに移動させる(ななめ移動可)
  - X、Yどちらか一方を優先させ、X(あるいはY)の座標が一致したら、他の座標を調べ方向を変える。
  - Xの距離とYの距離をくらべて、遠い方(あるいは近い方)を優先させる。
- などがあります。

どの方法を選ぶかは、そのゲームの内容にあわせて決めます。ここでは、プレイヤーの移動が、ななめ移動できないので、敵キャラクタもななめ移動はしないものとして、始めの方法ははずします。

二番目の方法は“迷路の移動”などで使われる場合がありますが、ここで設定した様なゲームでは動きがぎこちなく見えます。

そこで、3番目の方法を使ってみましょう。

また、常に追いかけてばかりいるのではプレイヤーが、かないませんから、追いかける割合を設定し、ランダムによって何回かに一回の割で、追いかける方向設定を行います。

さらに、何回かに一回の割で、方向をでたらめにセットし、相手を見失う様な効果も付け加えてみましょう。もちろん、障害物に当たった時は方向を変えるようにします。



リスト6-13 追いかけて移動プログラム

```

;
; MOVE CHR BY OIKAKE      XO=(E021H)      RO=(E029H)
;                          YO=(E022H)      OO=(E02AH)
;                          DO=(E023H)
;                          PO=(E024H)
;                          W01=(E027H)
;                          W02=(E028H)
;
; ORG 0BA00H
;
BA00 3A20E0  OAMOV: LD  A,(0E020H)      ;READ SET FLAG
BA03 A7      AND  A
BA04 200D    JR   NZ,OAMOV0
;
BA06 2120E0  LD   HL,0E020H
BA09 CDC0BB  CALL 0BBC0H      ;SET START POINT
BA0C 2120E0  LD   HL,0E020H
BA0F CD00BF  CALL 0BF00H      ;DRAW
BA12 C9      RET
;
BA13 3A27E0  OAMOV0:LD  A,(0E027H)      ;READ WAIT 1
BA16 A7      AND  A
BA17 2805    JR   Z,OAMOV1
BA19 3D      DEC  A      ;WAIT COUNT-1
BA1A 3227E0  LD   (0E027H),A
BA1D C9      RET
;
BA1E 3A28E0  OAMOV1:LD  A,(0E028H)      ;READ WAIT 2
BA21 3227E0  LD   (0E027H),A      ;SET WAIT COUNT
;
BA24 3A2AE0  LD   A,(0E02AH)      ;READ OIKAKE WARIAI
BA27 3C      INC  A
BA28 3C      INC  A
BA29 4F      LD   C,A
BA2A CD20BE  CALL 0BE20H
BA2D 0D      DEC  C
BA2E C2ADBA  JP   NZ,OIKAKE      ;OIKAKE
;
BA31 3A29E0  LD   A,(0E029H)      ;READ RAND NO.
BA34 3C      INC  A
BA35 3C      INC  A      ;MINMAM 2
BA36 4F      LD   C,A
BA37 CD20BE  CALL 0BE20H      ;RAND
BA3A 0D      DEC  C
BA3B 2805    JR   Z,OAMOV2
BA3D 3A23E0  LD   A,(0E023H)      ;READ DIR.
BA40 1809    JR   OAMOV4
;
BA42 0E04    OAMOV2:LD  C,4      ;RAND CHANGE
BA44 CD20BE  CALL 0BE20H
BA47 79      LD   A,C
BA48 3223E0  LD   (0E023H),A      ;SET DIR
;
BA4B 3A21E0  OAMOV4:LD  A,(0E021H)      ;READ XR
BA4E 5F      LD   E,A      ;E=X
BA4F 3A22E0  LD   A,(0E022H)      ;READ YR
BA52 57      LD   D,A      ;D=Y
BA53 3A23E0  LD   A,(0E023H)      ;READ DIR.

```



```

BA56 3D          DEC  A
BA57 2815        JR   Z,OAMR          ;GOTO RIGHT
BA59 3D          DEC  A
BA5A 280B        JR   Z,OAMU          ;GOTO UP
BA5C 3D          DEC  A
BA5D 2817        JR   Z,OAML          ;GOTO LEFT
;
BA5F 7A          LD   A,D             ;GOTO DOWN
BA60 FE2E        CP   46             ;DOWN LIMMIT 46
BA62 2840        JR   Z,OAMOFF
BA64 14          INC  D             ;Y=Y+1
BA65 1814        JR   OAMOV3         ;GOTO DRAW
;
BA67 7A          OAMU: LD   A,D             ;READ Y
BA68 A7          AND  A
BA69 2839        JR   Z,OAMOFF
BA6B 15          DEC  D             ;Y=Y-1
BA6C 180D        JR   OAMOV3         ;GOTO DRAW
;
BA6E 7B          OAMR: LD   A,E             ;READ X
BA6F FE4C        CP   76             ;RIGHT LIMMIT 76
BA71 2831        JR   Z,OAMOFF
BA73 1C          INC  E             ;X=X+1
BA74 1805        JR   OAMOV3         ;GOTO DRAW
;
BA76 7B          OAML: LD   A,E             ;READ X
BA77 A7          AND  A             ;LEFT LIMMIT 0
BA78 282A        JR   Z,OAMOFF
BA7A 1D          DEC  E             ;X=X-1
;
BA7B CDC0BD      OAMOV3:CALL 0BDC0H         ;CHECK FILED
BA7E 38C2        JR   C,OAMOV2
;
BA80 3A24E0      LD   A,(0E024H)         ;POSE 0,1 CHANGE
BA83 3D          DEC  A
BA84 2802        JR   Z,OAMOV5
BA86 3E01        LD   A,1
BA88 3224E0      OAMOV5:LD   (0E024H),A
;
BA8B 7B          LD   A,E
BA8C 3221E0      LD   (0E021H),A         ;SET X
BA8F 7A          LD   A,D
BA90 3222E0      LD   (0E022H),A         ;SET Y
BA93 2120E0      LD   HL,0E020H
BA96 E5          PUSH HL
BA97 E5          PUSH HL
BA98 CD50BF      CALL 0BF50H             ;CHR CLS
BA9B E1          POP  HL
BA9C CD00BF      CALL 0BF00H             ;CHR PUT
BA9F E1          POP  HL
BAA0 CD70BD      CALL 0BD70H             ;HANTEI
BAA3 C9          RET
;
BAA4 2120E0      OAMOFF:LD   HL,0E020H
BAA7 3600        LD   (HL),0             ;OFF
BAA9 CD50BF      CALL 0BF50H             ;CHR CLS
BAAC C9          RET
;
; OIKAKE

```



```

;
BAAD 3A21E0 OIKAKE:LD A,(0E021H) ;READ XO
BAB0 4F LD C,A
BAB1 3A01E0 LD A,(0E001H) ;READ XM
BAB4 5F LD E,A
BAB5 B9 CP C ;XM-XO
BAB6 3805 JR C,OIKAK1
BAB8 91 SUB C
BAB9 2E01 LD L,1 ;L=1 RIGHT
BABB 1804 JR OIKAK2
;
BABD 79 OIKAK1:LD A,C
BABE 93 SUB E ;XO-XM
BABF 2E03 LD L,3 ;L=3 LEFT
BAC1 4F OIKAK2:LD C,A ;C=ABS(XM-XO)
BAC2 3A22E0 LD A,(0E022H) ;READ YO
BAC5 47 LD B,A
BAC6 3A02E0 LD A,(0E002H) ;READ YM
BAC9 57 LD D,A
BACA B8 CP B ;YM-YO
BACB 3805 JR C,OIKAK3
BACD 90 SUB B
BACE 2600 LD H,0 ;H=0 DOWN
BAD0 1804 JR OIKAK4
;
BAD2 78 OIKAK3:LD A,B
BAD3 92 SUB D ;YO-YM
BAD4 2602 LD H,2 ;H=2 UP
BAD6 47 OIKAK4:LD B,A ;B=ABS(YM-YO)
BAD7 79 LD A,C
BAD8 B8 CP B ;DX-DY
BAD9 3803 JR C,OIKAK5
BADB 7D LD A,L ;IF DX>=DY
BADC 1801 JR OIKAK6
;
BADE 7C OIKAK5:LD A,H
BADF 3223E0 OIKAK6:LD (0E023H),A ;SET DIR
;
BAE2 C34BBA JP OAMOV4 ;GOTO MOVE
;
;

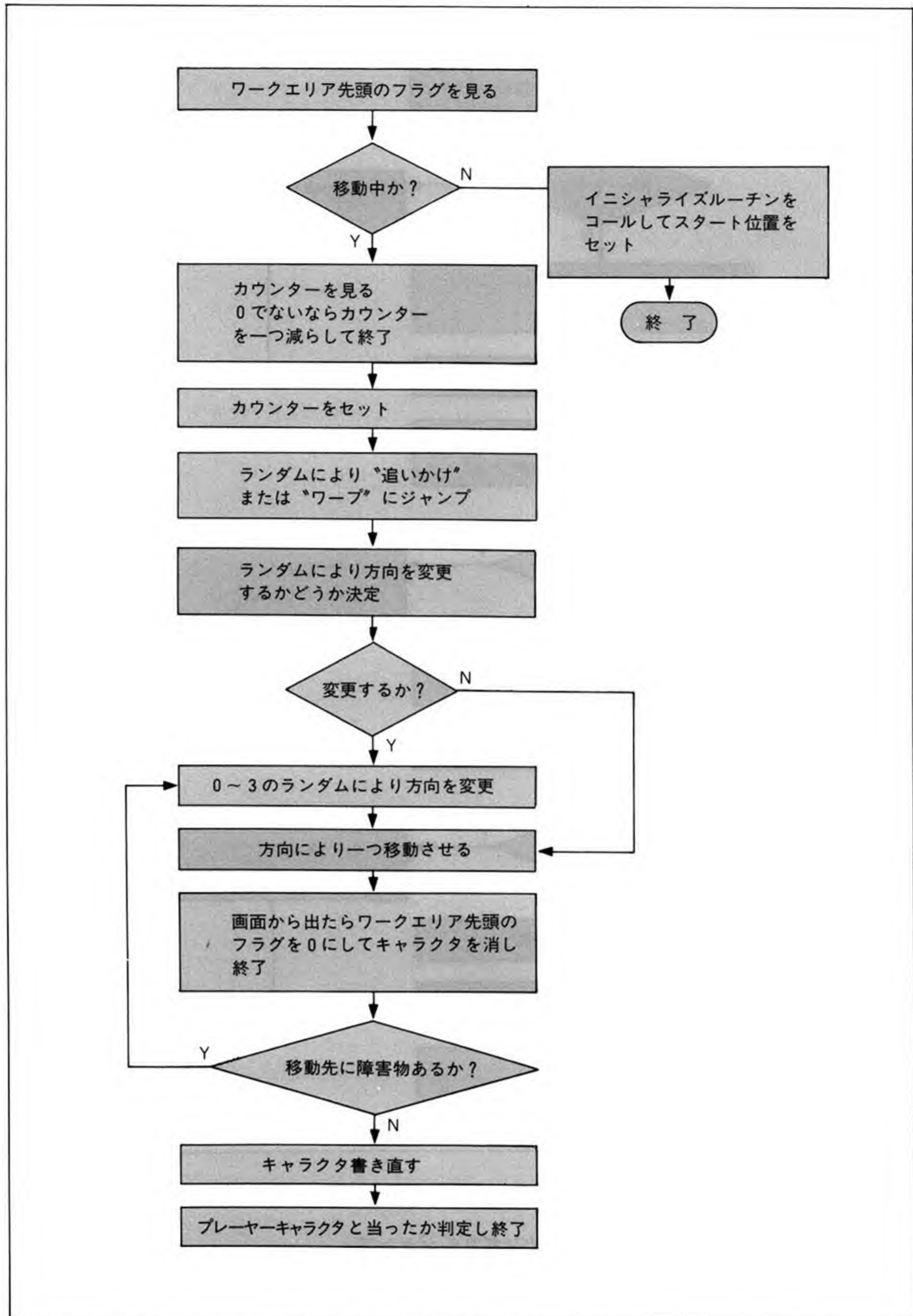
```

具体的なプログラムをリスト6-13にのせておきます。方向による移動部分は、今までのものとそっくり同じです。データによる移動ルーチンとの違いは、追いかける所をのぞくと、データによる方向変化のかわりに、ランダム方向変化を付け加え、障害物に当たった時も、ここにジャンプさせています。第6-9図が大体の作業手順です。

第6-10図は、追いかける方向変更の部分をフローチャートにしたもので、少しいねいに書いてみました。

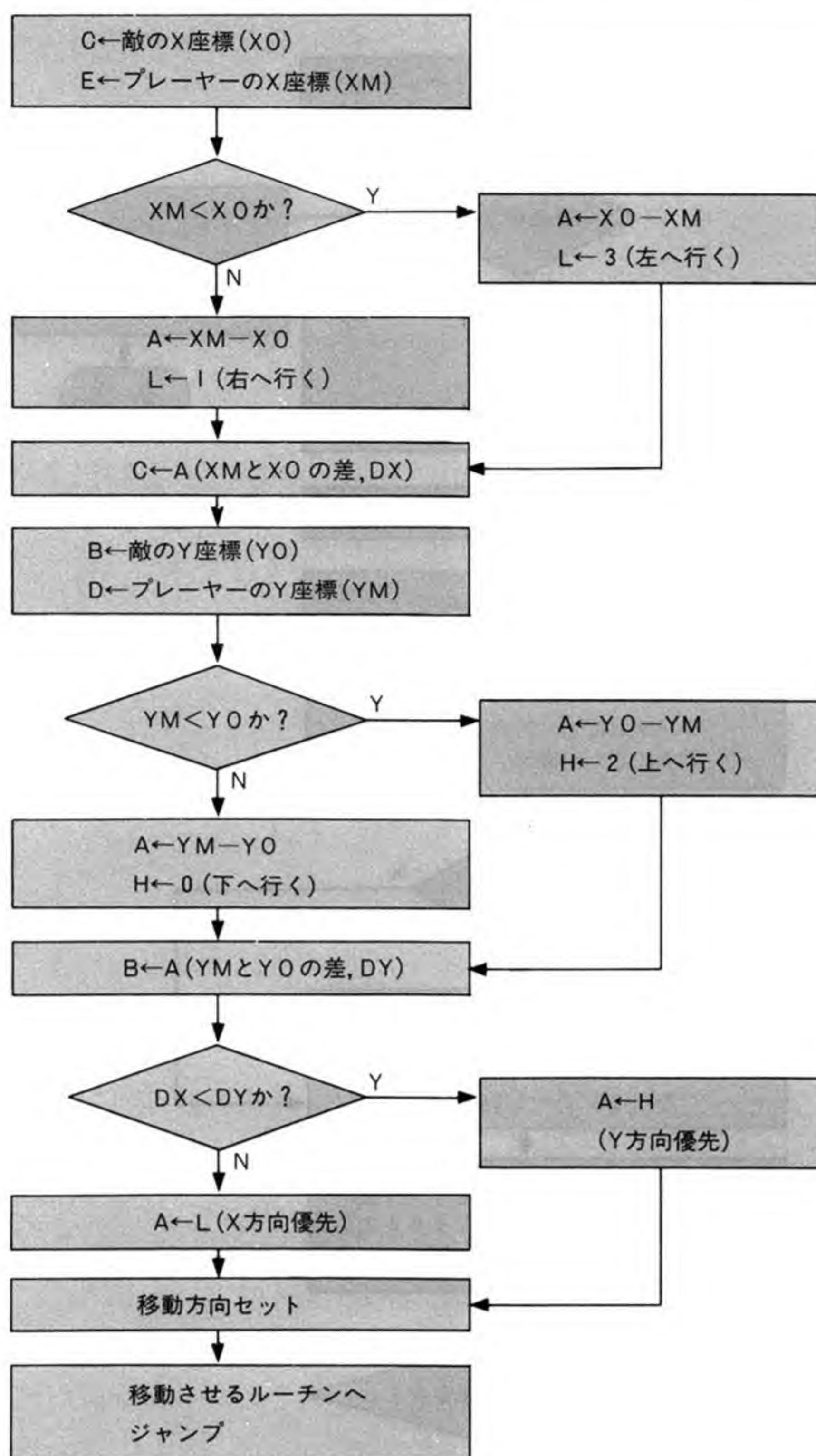
追いかける割り合いを決めるのは、ワークエリアの&HE02Aに入っている数を見ます。この数が多い程、追いかける割合が大きくなりますが、“0”でもかなり追いかける割合が大きいので、&HBA2Eの“C2”を“CA”にかえると、&HE02Aの数が多い程、追いかける割合が大きくなります。この方がよいかし





第6-9図 追いかけるおよびワープ移動の共通部分作業手順





第6-10図 追いかける手順

れません。C2とCAの違いは次の様です。

C2 ADBA JP NZ, 0BAADH;ノンゼロでジャンプ

CA ADBA JP Z, 0BAADH;ゼロでジャンプ

乱数による方向変更は、ワークエリアの&HE 029の値を見て変更する割合を決めています。この値が大きい程、ランダム変更をしなくなります。“追いかけ”と同じ様に、&HE 029の値に2を足し、元の値が0でも2になるようにしてあります。それを乱数ルーチンにかけて、戻って来



た数が"1"の時だけランダム変更させるのです。

&HE 029と&HE 02Aの値はあらかじめ書き込んでおきます。

このプログラムの実験に移る前に、もう一つオマケに、"ワープ移動"を作ってみましょう。リスト6-14がそのプログラムです。メインになる部分は、リスト6-13とほとんど同じですね。ワークエリアが前のプログラムは&HE 020からだったのに対し、&HE 030から使っている事と、"追いかけるルーチン"へジャンプするかわりに、20回に1回の割りで"ワープルーチン"へジャンプさせて

#### リスト6-14 ワープ移動プログラム

```

;
; MOVE CHR BY RAND      XR=(E031H)
;                        YR=(E032H)
;                        DR=(E033H)
;                        PR=(E034H)
;                        WR1=(E037H)
;                        WR2=(E038H)
;
;                        ORG 0B900H
;
B900 3A30E0  RAMOV: LD  A,(0E030H)      ;READ SET FLAG
B903 A7      AND  A
B904 200D    JR   NZ,RAMOV0
;
B906 2130E0  LD   HL,0E030H
B909 CDC0BB  CALL 0BBC0H      ;SET START POINT
B90C 2130E0  LD   HL,0E030H
B90F CD00BF  CALL 0BF00H      ;DRAW
B912 C9      RET
;
B913 3A37E0  RAMOV0:LD  A,(0E037H)    ;READ WAIT 1
B916 A7      AND  A
B917 2805    JR   Z,RAMOV1
B919 3D      DEC  A                ;WAIT COUNT-1
B91A 3237E0  LD   (0E037H),A
B91D C9      RET
;
B91E 3A38E0  RAMOV1:LD  A,(0E038H)    ;READ WAIT 2
B921 3237E0  LD   (0E037H),A          ;SET WAIT COUNT
;
B924 0E14    LD   C,20
B926 CD20BE  CALL 0BE20H      ;RAND(20)
B929 0D      DEC  C
B92A CAA9B9  JP   Z,RANWAP      ;RANDOM 7-7°
;
B92D 3A39E0  LD   A,(0E039H)    ;READ RAND NO.
B930 3C      INC  A
B931 3C      INC  A                ;MINMAM 2
B932 4F      LD   C,A
B933 CD20BE  CALL 0BE20H      ;RAND
B936 0D      DEC  C
B937 2805    JR   Z,RAMOV2
B939 3A33E0  LD   A,(0E033H)    ;READ DIR.
B93C 1809    JR   RAMOV4

```



```

;
B93E 0E04      RAMOV2:LD    C,4                ;RAND CHANGE
B940 CD20BE          CALL 0BE20H
B943 79          LD    A,C
B944 3233E0      LD    (0E033H),A            ;SET DIR
;
B947 3A31E0      RAMOV4:LD    A,(0E031H)      ;READ XR
B94A 5F          LD    E,A                  ;E=X
B94B 3A32E0      LD    A,(0E032H)            ;READ YR
B94E 57          LD    D,A                  ;D=Y
B94F 3A33E0      LD    A,(0E033H)            ;READ DIR.
B952 3D          DEC    A
B953 2815        JR    Z,RAMR                ;GOTO RIGHT
B955 3D          DEC    A
B956 280B        JR    Z,RAMU                ;GOTO UP
B958 3D          DEC    A
B959 2817        JR    Z,RAML                ;GOTO LEFT
;
B95B 7A          LD    A,D                  ;GOTO DOWN
B95C FE2E        CP    46                  ;DOWN LIMMIT 46
B95E 2840        JR    Z,RAMOFF
B960 14          INC    D                  ;Y=Y+1
B961 1814        JR    RAMOV3                ;GOTO DRAW
;
B963 7A          RAMU: LD    A,D              ;READ Y
B964 A7          AND    A
B965 2839        JR    Z,RAMOFF
B967 15          DEC    D                  ;Y=Y-1
B968 180D        JR    RAMOV3                ;GOTO DRAW
;
B96A 7B          RAMR: LD    A,E              ;READ X
B96B FE4C        CP    76                  ;RIGHT LIMMIT 76
B96D 2831        JR    Z,RAMOFF
B96F 1C          INC    E                  ;X=X+1
B970 1805        JR    RAMOV3                ;GOTO DRAW
;
B972 7B          RAML: LD    A,E              ;READ X
B973 A7          AND    A                  ;LEFT LIMMIT 0
B974 282A        JR    Z,RAMOFF
B976 1D          DEC    E                  ;X=X-1
;
B977 CDC0BD      RAMOV3:CALL 0BDC0H           ;CHECK FILED
B97A 38C2        JR    C,RAMOV2
;
B97C 3A34E0      LD    A,(0E034H)           ;POSE 0,1 CHANGE
B97F 3D          DEC    A
B980 2802        JR    Z,RAMOV5
B982 3E01        LD    A,1
B984 3234E0      RAMOV5:LD    (0E034H),A
;
B987 7B          LD    A,E
B988 3231E0      LD    (0E031H),A           ;SET X
B98B 7A          LD    A,D
B98C 3232E0      LD    (0E032H),A           ;SET Y
B98F 2130E0      LD    HL,0E030H
B992 E5          PUSH HL
B993 E5          PUSH HL

```



```

B994 CD50BF      CALL 0BF50H      ;CHR CLS
B997 E1          POP HL
B998 CD00BF      CALL 0BF00H      ;CHR PUT
B99B E1          POP HL
B99C CD70BD      CALL 0BD70H      ;HANTEI
B99F C9          RET

;
B9A0 2130E0      RAMOFF:LD HL,0E030H
B9A3 3600        LD (HL),0      ;OFF
B9A5 CD50BF      CALL 0BF50H      ;CHR CLS
B9A8 C9          RET

;
; 7-7°
;
B9A9 0E28        RANWAP:LD C,40
B9AB CD20BE      CALL 0BE20H
B9AE 3E12        LD A,18
B9B0 81          ADD A,C      ;X=18+RAND(40)
B9B1 5F          LD E,A      ;E=XR
;
B9B2 0E1E        LD C,30
B9B4 CD20BE      CALL 0BE20H
B9B7 3E08        LD A,8
B9B9 81          ADD A,C      ;Y=8+RAND(30)
B9BA 57          LD D,A      ;D=YR
B9BB D5          PUSH DE
B9BC CDC0BD      CALL 0BDC0H      ;FILED CHECK
B9BF D1          POP DE
B9C0 38E7        JR C,RANWAP
;
B9C2 2130E0      LD HL,0E030H
B9C5 D5          PUSH DE
B9C6 CD70BD      CALL 0BD70H      ;ATTACK CHECK
B9C9 D1          POP DE
B9CA 38DD        JR C,RANWAP
;
B9CC 7B          LD A,E
B9CD 3231E0      LD (0E031H),A      ;SET X
B9D0 7A          LD A,D
B9D1 3232E0      LD (0E032H),A      ;SET Y
B9D4 0E04        LD C,4      ;RAND CHANGE
B9D6 CD20BE      CALL 0BE20H
B9D9 79          LD A,C
B9DA 3233E0      LD (0E033H),A      ;SET DIR
B9DD 2130E0      LD HL,0E030H
B9E0 E5          PUSH HL
B9E1 CD50BF      CALL 0BF50H      ;CHR CLS
B9E4 E1          POP HL
B9E5 CD00BF      CALL 0BF00H      ;CHR PUT
B9E8 C9          RET
;
;

```



います。ワークエリアについては、第6-2図を見直して下さい。

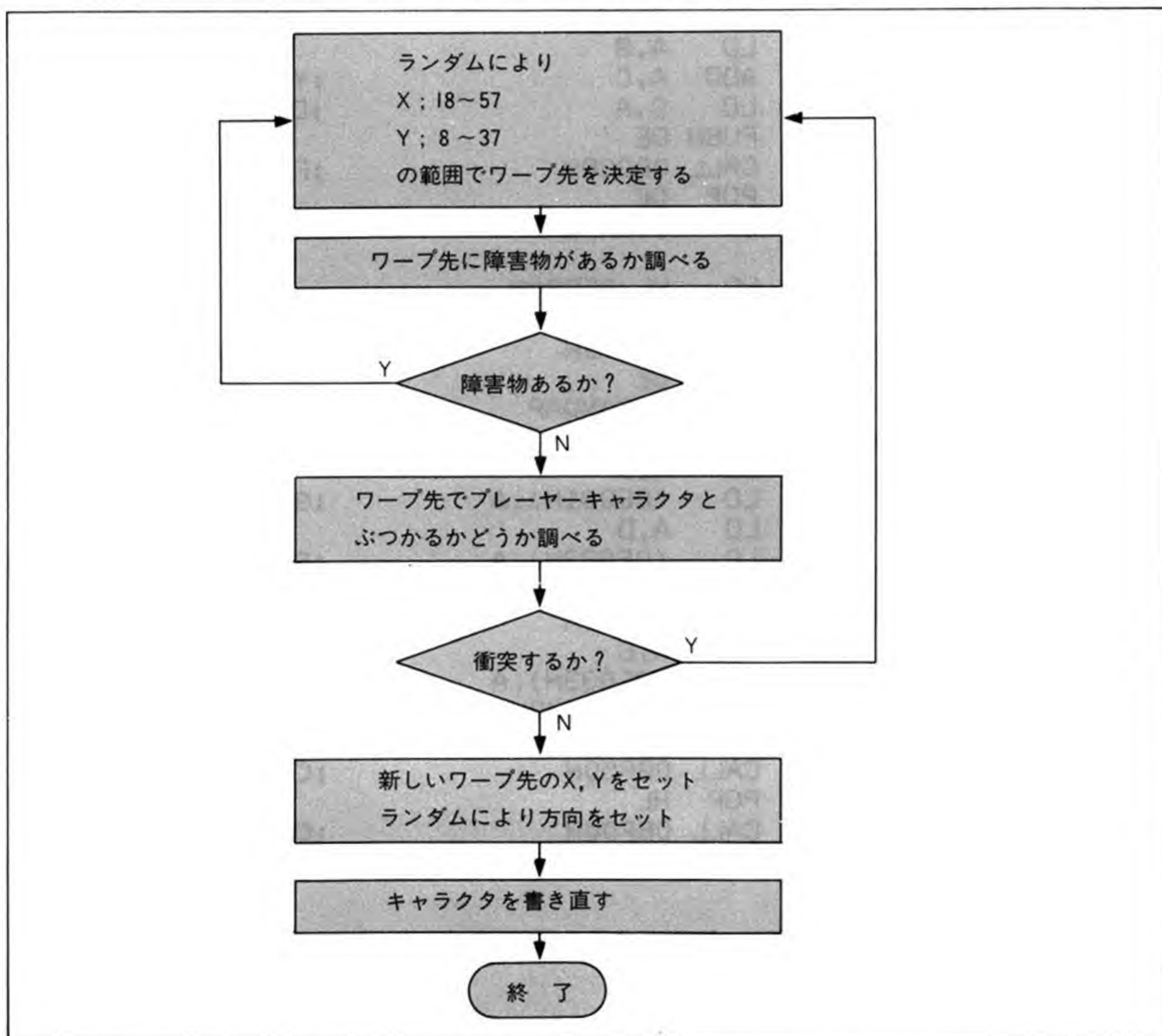
新しく登場したのはワープルーチンですが、難しい事は何もありません。第6-11図がその手順です。

やっている事は、ランダムによりワープ先の座標を決め、その場所に障害物や、プレイヤーのキャラクターがいたら、ワープ先を決め直します。いきなり自分のキャラクターのいる所に敵がワープしてやられたのでは、誰でも腹が立つでしょうから、そうならないようにしました。ワープ可能ならば、方向をでたらめにセットして、古いキャラクターを消し、新しいキャラクターをワープ場所にセットします。

まだまだアイデア次第で、様々な移動のし方ができると思いますが、この辺で切り上げ、サブルーチンをまとめて実験してみましょう。

まず、今までに使って来た、キャラクターデータや、仮想空間のフィールド(マップ)データ、それに、必要なプログラムをすべてロードします。データ移動はフィールドの障害物を消してしまいますので、扱いませんが、消えてもかまわない方はロードして下さい。

ワークエリアの&HE 000から&HE 03Fを0で埋めたあと、&HE 029, &HE 02A, &HE 039に入



第6-11図 ワープさせる手順







部分を0でつぶします。

リスト6-16は、このマシン語プログラムとリンクさせたBASICプログラムです。ワークエリアの初期化も行っています。

以上、一つのゲームを例に、いくつかのルーチンを組んでみました。色々と変えて実験してみてください。

一つの目的のプログラムでも、いく通りもの作り方があります。例に上げたものも書き替えによって処理速度が上がるなど、より良くなる所もありますが、まず、完全に動くものを作り、マシン語に慣れるうちにさらに良い物が作れるようになると思います。あせらず、少しずつ取り組んで下さい。

ゲーム作りの第一歩は、キャラクタのPUTだと考えますので、ここにもう一度、グラフィックRAMをアクセスする上での注意点を上げておきましょう。

1. グラフィックRAMをアクセスする前に、“割り込み”を禁止する。割り込み処理のテーブルがグラフィックRAMの裏のメインRAM上にあります。
2. スタックポインタをグラフィックRAMのアドレス、&HC000より前に移動させる。グラフィックRAMをアクセスしている時でも“CALL”や“PUSH・POP”が使えるようにします。
3. グラフィック表示に限らず、PUSH・POPに気をつける。PUSHしたら必ずPOPするように、たとえば、PUSHしたままリターンすると、おかしい所へプログラムカウンタが飛んでしまいます。

#### リスト6-16 BASIC MAIN

```

100 CLEAR , &HB7FF:CONSOLE 0, 25, 0, 1
110 GAM=&HB800:P=&HE009:R1=&HE029:R2=&HE039:O=&HE02A
120 WM=&HE008:WD=&HE018:WO=&HE028:WW=&HE038
130 MX=&HE001:MY=&HE002
140 FOR I=&HE000 TO &HE03F:POKE I, 0:NEXT
150 WMD=0          : ' MY CHR WAIT 0
160 WDD=1          : ' DATA MOVE CHR WAIT 1
170 WOD=3          : ' OIKAKE CHR WAIT 3
180 WWD=2          : ' WARP CHR WAIT 2
190 R1D=50         : ' OIKAKE RAND CHANGE WARIAI
200 R2D=80         : ' WARP RAND CHANGE WARIAI
210 OD=1           : ' OIKAKE WARIAI
220 MXD=40         : ' START MY X
230 MYD=23         : ' START MY Y
240 POKE WM, WMD:POKE WD, WDD:POKE WO, WOD:POKE WW, WWD
250 POKE R1, R1D:POKE R2, R2D:POKE O, OD
260 POKE MX, MXD:POKE MY, MYD
270 CLS
280 OUT &HE6, 0:CALL GAM:OUT &HE6, 6
290 CLS 2:LOCATE 10, 10:PRINT " POINT = ";PEEK(P)
300 END

```



よく、マシン語で暴走して、リセットさせてメモリを見てみると、同じ数の繰り返しになっている事がありますが、これなどは大抵、PUSH・POPのミスによるものです(経験上……)。

## 6-5

### スクロール画面を作るには

これからダウンスクロールの方法を二通り紹介いたします。

まず、“GAMMA 5”の様なスクロールです。一つのスクロール画面全てを下にずらして書き直すが、スピード上問題となる場合に有効で、スクロール画面のうち、自立つキャラクタのみを移動させて行くと言うものです。

この方法を私達は“キャラクタスクロール”と呼びます。具体的なやり方は次の通りです。

1. 全体のスクロール画面を、ある大きさ(GAMMA5の場合は横3バイトに縦12ライン)のキャラクタの集合によるものとする。
2. 構成するキャラクタを、**スタート時のみ書くキャラクタ**と、**スクロール中も書くキャラクタ**に分ける。
3. スクロール中も書くキャラクタは、キャラクタの上2ラインを地表を表すキャラクタ等と同じにして、2ライン下げてプットした時、前のキャラクタを消すと同時に、**わだちとして地表も書いてしまうようにする**。つまり、書くキャラクタを2ラインずつ下げてプットする事により、移動した跡が、**地表や縦のライン、横の壁などになる所がミソです(第6-12図)。**

この方法の利点は、書くキャラクタが少なければ、かなりの高速スクロールが可能と言う点ですが、欠点は、どうしても書くキャラクタ数にムラが生じるので、スクロールスピードが一定しないと言う点です。

これを補うために、書くキャラクタ数をカウントして、数により時間待ちのタイマーを入れてやるようにします。

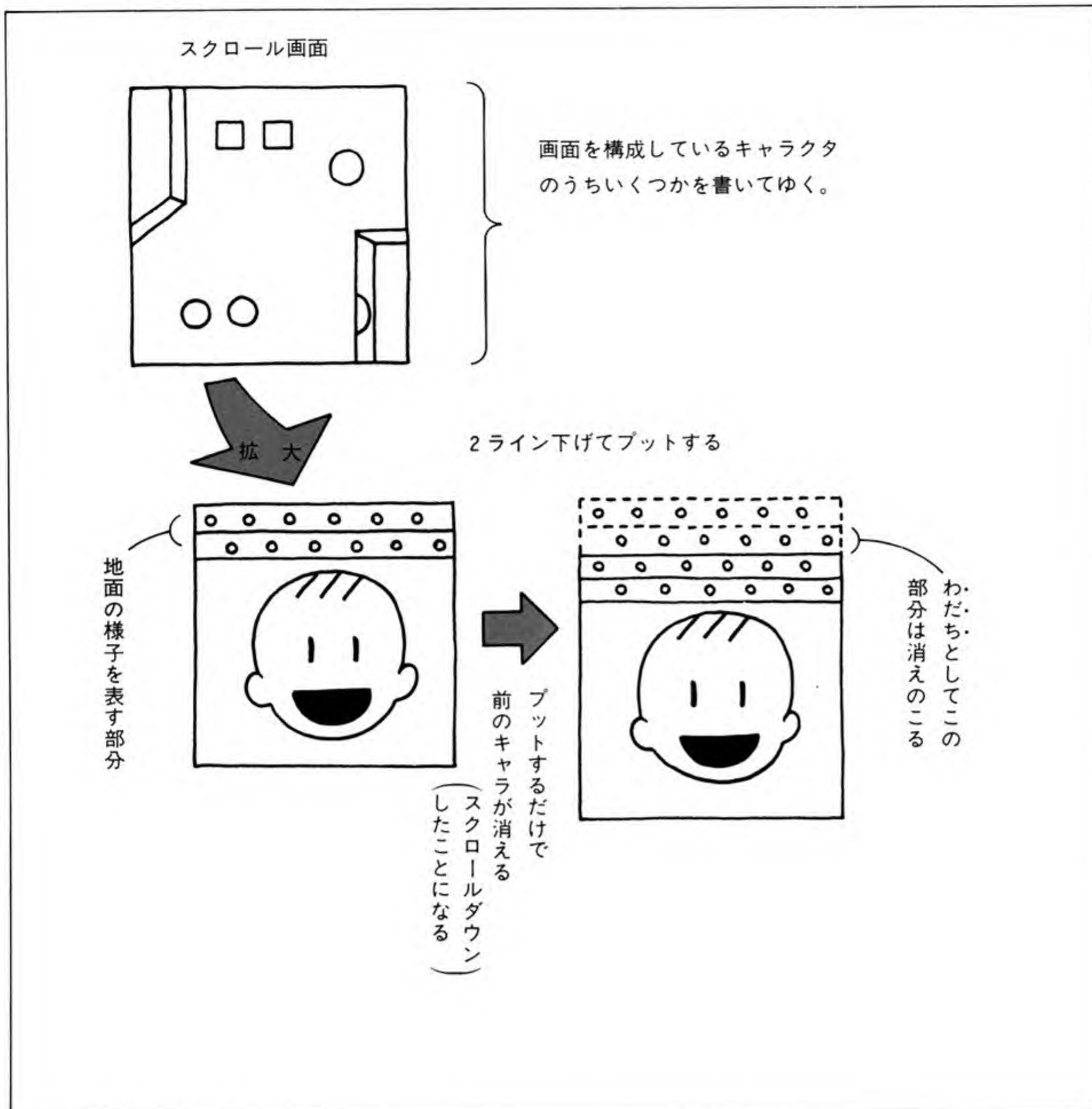
また、キャラクタをプットする時に注意しなければならないのは、画面上部と、画面下部では、一つのキャラクタの上、あるいは下の部分が、画面の外になってしまう事です。その部分のプットルーチンを工夫しなければなりません。

“GAMMA 5”の場合、スクロールに必要な変数は二つだけで、“スクロール画面先頭のマップデータアドレス”と、“1キャラクタ12ラインのうち何ラインずらすか”の二つです。前の変数名を“MAD”後の変数名を“LIN”とすると、スクロール作業の手順は、ざっと次のようになります。

1. MADにスタートの時の画面先頭マップデータアドレスをセットし、マップデータを見ながら、そのデータに相当するキャラクタを、一画面分すべてプットする。LINは0にイニシャライズしておく。
2. LINの値を2ふやす。もし、LINの値が12になってしまったら、LINを0に戻し、MADの値







第6-12図 キャラクタスクロールのしくみ

を16少なくする。つまり、キャラクター一列が16個なので、一列分前のデータアドレスにする。

3. MADで示されるマップデータの値を一画面分順に見て行き、スクロール中に書くキャラクターのみプットする。プットする場所は、LINで示される値の分だけ基準の場所より何ラインか下げる。

4. MADの値が決められた値(マップデータの終り)になるまで、2に戻ってくり返す。

以上です。

次に、スクロール画面をすべて書き替えてスクロールさせる方法を考えてみましょう。

やり方としてはこちらの方が単純で、ダウンスクロールの場合ならば、スクロール画面にあるデータをすべて何ラインか下げて転写します。一画面スクロールさせ、上の部分に新しいデータを書き込めばでき上りです。

気をつけなければいけないのは、スクロール画面上方から転写作業を行うと、同じ画面データが下へ下へと転写されるために、スクロールできません。画面一番下のデータから、転写作業を




行うようにします。

このやり方の利点は、スクロール速度が一定であること、プログラムが単純であること、キャラクタの大きさにこだわる必要がないこと、などが上げられます。欠点は何と言っても、一つのスクロール画面を転写するのに時間がかかりすぎるのですが、PC-88SR以降、3画面同時書き込みモードなどにより、処理速度が格段に上っていますので、全部転写してもさほど時間がかからないようになりました。


リスト6-17に、画面左半分を2ラインずつダウンスクロールさせるプログラムをのせておきますので、そのスピードを目で確かめて下さい。

このプログラムをテストするには、

CLEAR, &HBEFF 


などとして、スタックエリアを移動させてから、画面に、適当な絵を表示させておきます。

たとえば、

CIRCLE(320, 100), 200, 7 

とすると、画面一杯に円が描かれるでしょう。

そして、モニタから、

h)GBF00 

とやってみると、円の左半分が下にスクロールし、下へ消えた部分が上から出現して、また元の円に戻るでしょう。下へ消えた部分のデータは、一旦、グラフィックRAMの画面に表示されない所、&HFE80からに保存しておき、一画面転写した後、画面上部2ラインにプットします。

リスト6-17 2ラインスクロール

```

;
; ALU SCROLL
;
;
;          ORG  0BF00H
;
BF00 F3      SCROLL:DI                      ;ワリコミ タメ
;
BF01 DB32    IN   A,(32H)
BF03 F5      PUSH AF
BF04 F640    OR   40H                      ;KAKUTHIOO MODE
BF06 D332    OUT  (32H),A
;
BF08 3E90    LD   A,90H
BF0A D335    OUT  (35H),A                  ;READ & WRITE
BF0C 3E07    LD   A,7
BF0E D334    OUT  (34H),A                  ;BIT SET MODE
;
BF10 0664    LD   B,100                    ;100 カイ
BF12 C5      SCROL1:PUSH BC
;
BF13 21E0FD  LD   HL,0FDE0H                ;198 LINE TOP ADDRESS
BF16 E5      PUSH HL
BF17 11A000  LD   DE,0A0H
BF1A 19      ADD  HL,DE
BF1B EB      EX   DE,HL                    ;DE=HL+ 2 LINES
BF1C E1      POP  HL
;

```



```

BF1D 0664          LD    B,100          ;1 カメン フン
BF1F C5            SCROL2:PUSH BC
BF20 E5            PUSH HL
BF21 E5            PUSH HL
BF22 012800        LD    BC,28H          ;HALF 1 LINE BYTES
BF25 CD4CBF        CALL SCROLS          ;2 LINES フロック テンソク SUB
BF28 E1            POP  HL
BF29 11A000        LD    DE,0A0H        ;2 LINE BYTES
BF2C A7            AND   A
BF2D ED52          SBC   HL,DE          ;2 LINE UP
BF2F D1            POP  DE              ;DE=マイノ HL
BF30 C1            POP  BC
BF31 10EC          DJNZ SCROL2

;
BF33 1100C0        LD    DE,0C000H      ;SCREEN TOP
BF36 2180FE        LD    HL,0FE80H      ;SCREEN BOTTOM+1
BF39 012800        LD    BC,28H          ;HALF 1 LINE BYTES
BF3C CD4CBF        CALL SCROLS          ;2 LINES フロック テンソク SUB

;
BF3F C1            POP  BC
BF40 10D0          DJNZ SCROL1

;
BF42 AF            XOR   A
BF43 D335          OUT   (35H),A          ;MAIN RAM
BF45 F1            POP  AF              ;NORMAL MODE
BF46 D332          OUT   (32H),A
BF48 D35F          OUT   (5FH),A          ;RAM

;
BF4A FB            EI                  ;フリコミ OK!
BF4B C9            RET

;
BF4C C5            SCROLS:PUSH BC        ;2 LINE テンソク
BF4D E5            PUSH HL
BF4E D5            PUSH DE
BF4F EDB0          LDIR                  ;フロック テンソク (DE)=(HL) INC DE,INC H
BF51 E1            POP  HL              ;HL=DE
BF52 015000        LD    BC,50H
BF55 09            ADD   HL,BC
BF56 EB            EX    DE,HL          ;DE 1 LINE DOWN
BF57 E1            POP  HL
BF58 09            ADD   HL,BC          ;HL 1 LINE DOWN
BF59 C1            POP  BC
BF5A EDB0          LDIR                  ;フロック テンソク (DE)=(HL) INC DE,INC H
BF5C C9            RET

;
;

```

## 6-6

### まとめ

最後に、ハンドアセンブルでゲームを作るまでの流れをまとめておきます。

1. どんなゲームを作るか考える。画面の様子を紙に書いてみる。ゲームの機能をメモする。さらに、一つ一つの動きに注目し、何が、どこで、どんな動きをするかおさえる。それらをまとめて表などにしておく。
2. アイデアがまとまったら、必要なキャラクタを書き出し、キャラクタデータを作る。メモリの事も少しは考えて多すぎないようにする。
3. メモリの割り付けをする。プログラムエリア、キャラクタデータエリア、マップデータエリア、ワークエリア、その他のデータエリア等、どこに何を置くかはっきりさせ、一覧表にまとめる。BGMをつけるなら、もちろん、そのエリアも確保しなければならない。
4. ワークエリアを作る。具体的に変数を決めるためには、一つ一つの移動の仕方等をはっきり



とさせなければならない。細かいフローチャートまでは必要なくとも、プログラムの手順ぐらいいはメモした方がよい。ワークエリアは、モニタによってすぐ見える位置がよく(テキストウィンドウを使わなければ見えない前半のメモリはよくない)、ディスクから立ち上げた時に、IPL(&HC000~&HC1FF)や、ディスクBASIC(&H8400~&HB3FF)でつぶされる所はさける。

また、実験したルーチンは、レジスタにある値を入れてコールする、と言う形のものが多かったが、始めのうちは、面倒でもできるだけワークエリアを通して数値のやりとりを行わせる、ようにした方が、デバッグしやすいと思います。

5. 表示ルーチンなどの基本的なサブルーチンを作り、動作を確認しておく。
6. 移動データや、仮想空間のマップデータが必要ならば、テストプレイ用のデータを作っておく。  
"データメイカー"等のツールを自作するのがよい。
7. 一種類ずつ、動作を見ながらプログラミングして行く。4で書いたように、ワークエリアの数値のみによりコントロールされるようにするとよい。テストプレイ用のメインプログラムは、"ESC"キーなどで中断がきくようにし、動作のおかしい時はまずワークエリアの数値が正常かチェックする。プログラムのどこが変か見つけやすい。

もし、決められた値以外の数値をとる変数があれば、一動作ごとに、VRAMにその数値を送ってやり、テキスト画面のキャラクタによってモニタするなどの手段もある。

8. 全体を組み合わせるメインルーチンを作る。メインルーチンは、確実に動作するサブルーチンを次々にコールして行くようになる。できるだけ単純なメインルーチンがよいと思う。テストプレイを行い、"判定"等がうまく行っているかチェックする。
9. 全体のバランスを考え、移動データやPUTデータを完全なものにする。スピードや攻撃の激しさなどを調整する。
10. 効果や、BGMは、やはりそれのみでテストプレイしておき、必要なプログラムからコール、または割り込ませる。

市販のソフトでは、タイトルやデモも重要になってきますが、以上のようなステップで作って行きます。

何と言ってもハンドアセンブルではモニタだけがたよりですから、プログラムやデータが、モニタを見てどこに何があるかわかる必要があります。ですから、"追いかけてこゲーム"の実験で使ったプログラムのように、**きりの良い番地から書き始める**ようにし、プログラムとプログラムの間は空白、(0)にして、**後から書き足してもよいようなスペースをとっておきましょう。**

また、モニタにはMコマンドでプログラムを複写する能力がありますから、リスト6-13とリスト6-14のように、少し変えれば同じプログラムと言う場合は、Mコマンドで複写してから必要な所を書き替えてやれば効率的です。

ハンドアセンブルによって作ったゲーム"ZENON"のリストは、レポート用紙2冊分ですが、複写して使った所もあり、実際にはそれより多くのメモリを使っています。

プログラムの量が多くなっても、一つ一つのサブルーチンが間違いなく動作すれば、そう簡単にバグは出ないものです。ぜひ長いプログラムにも挑戦してみてください。



「それでは、これからのパソコンについて、発明家のK氏にうかがってみる事にしましょう」

「はいはいはい。私が思いますにですね、パソコンはその使用目的にあわせて“分化”すると思うわけです、はい。つまり、ホビーはホビー、ビジネスはビジネスにと専用機になります。しかし、どの機械も使い方、入力の方法などは、きわめて簡単になるのは当然でしょう。キーボードは無くなるでしょうね、やはり」

「で、Kさん、今日はパソコンの未来を占う、様々な発明品を持って来ていただいているようですが、何から見せていただけますか？」

「例えばこれ。ヘッドホンステレオに見えますが、これがパーソナルコンピュータの一つの姿なのであります。これに個人の情報、スケジュールなどを入れておきますと、次に何をすればいいのか、音声で指示されます。さらに小型化すると、こうなります」

「磁気バンソウコウみたいですね」

「そうです。この位になると、こめかみに張り付ける“外付け型”と、直接脳に埋め込む“内蔵型”が考えられますね。もっともこれは本物じゃありませんよ。こんなのまだ作れっこない。ははは」

「はは、そりゃそうですね、安心しました。で、他には」

「では、私の傑作、ホビースペースパソコンを見ていただきたい。これがそうです」

「はあ、ちょっと見た所、AVステレオに見えますが……」

「AVステレオを改造したものですから当然です。しかし、これはアンプではなく、高性能コンピュータであり、上に載っているのはコンパクトディスクプレーヤーですが、これにプログラムを書き込んだCDを入れるわけです。フロッピーディスクよりも格段のデータが、CDでは記録できる事は御存知でしょう」

「は、はい。では動かしてみただけですか」

「これが、ゲームソフトのCDです。こいつを入れてスイッチを入れると……」

「ほおー、すごくきれいな絵と音ですね。しかもすごく速い——」

「トオーゼンです、ははは。で、操作ですが、これは本体内蔵のマイクを使います。音声入力です。すごいでしょ」

「ちょっとやってみて下さい」

「始める時はまず“スワート”と叫びます」

「わっ、びっくりした。いきなり大きな声出さないで下さいよ」

「ははは、でも、キャラクタが動き出したでしょ。昔なつかしい“すぺえすはりあ”ですよ。動きも、もちろん声で指示します。そらっ、右っ！、下っ！、右っ！ あっ、逃げろっ。——い、いや、“逃げろ”と言う命令はありませんでした。上っ、左っ、もっ、と左。下っ、撃て、撃て、撃て、右っ！。下だこら下っ！、撃て撃て、撃てって言ったこの馬鹿野郎——っ！！！！」

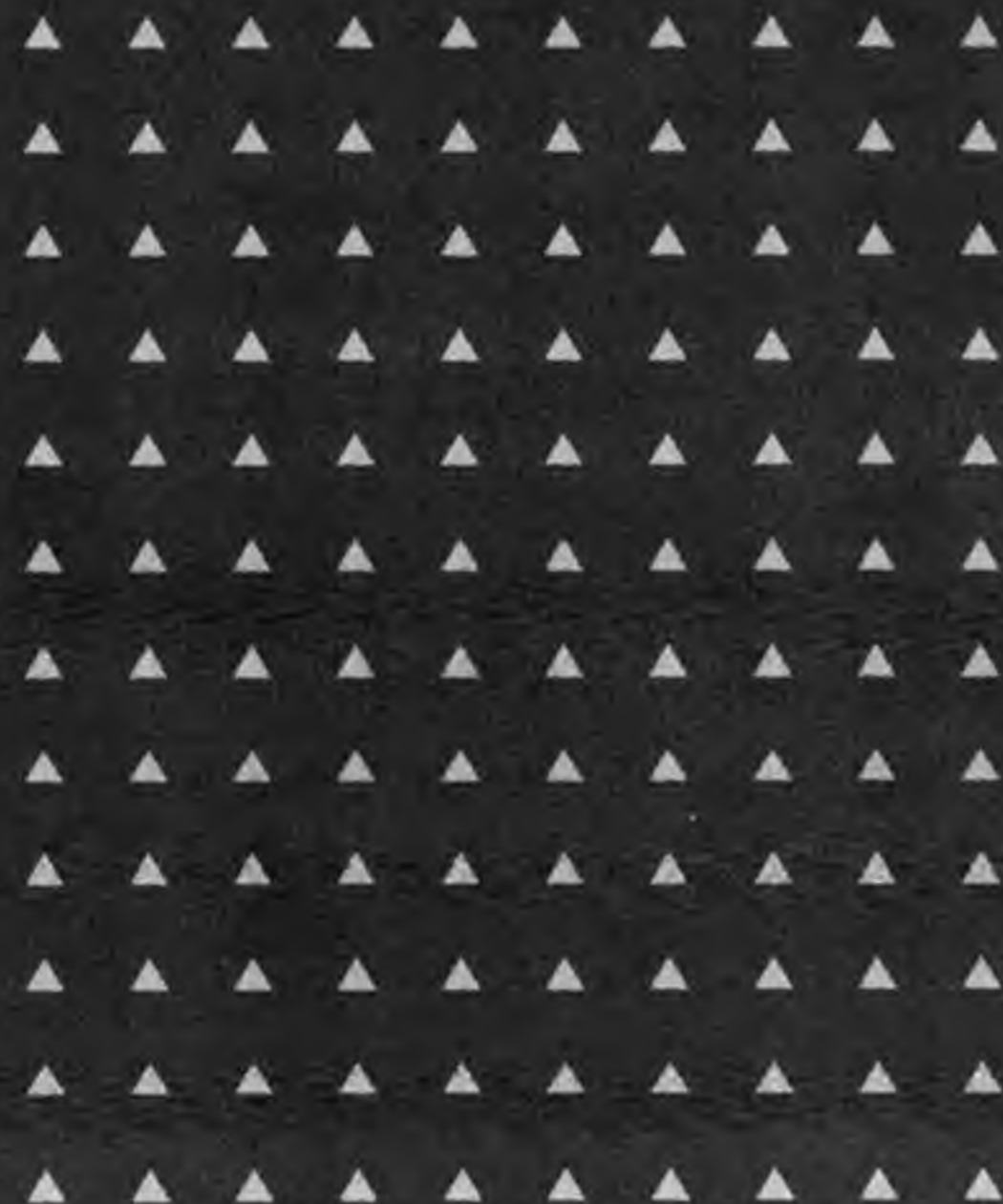
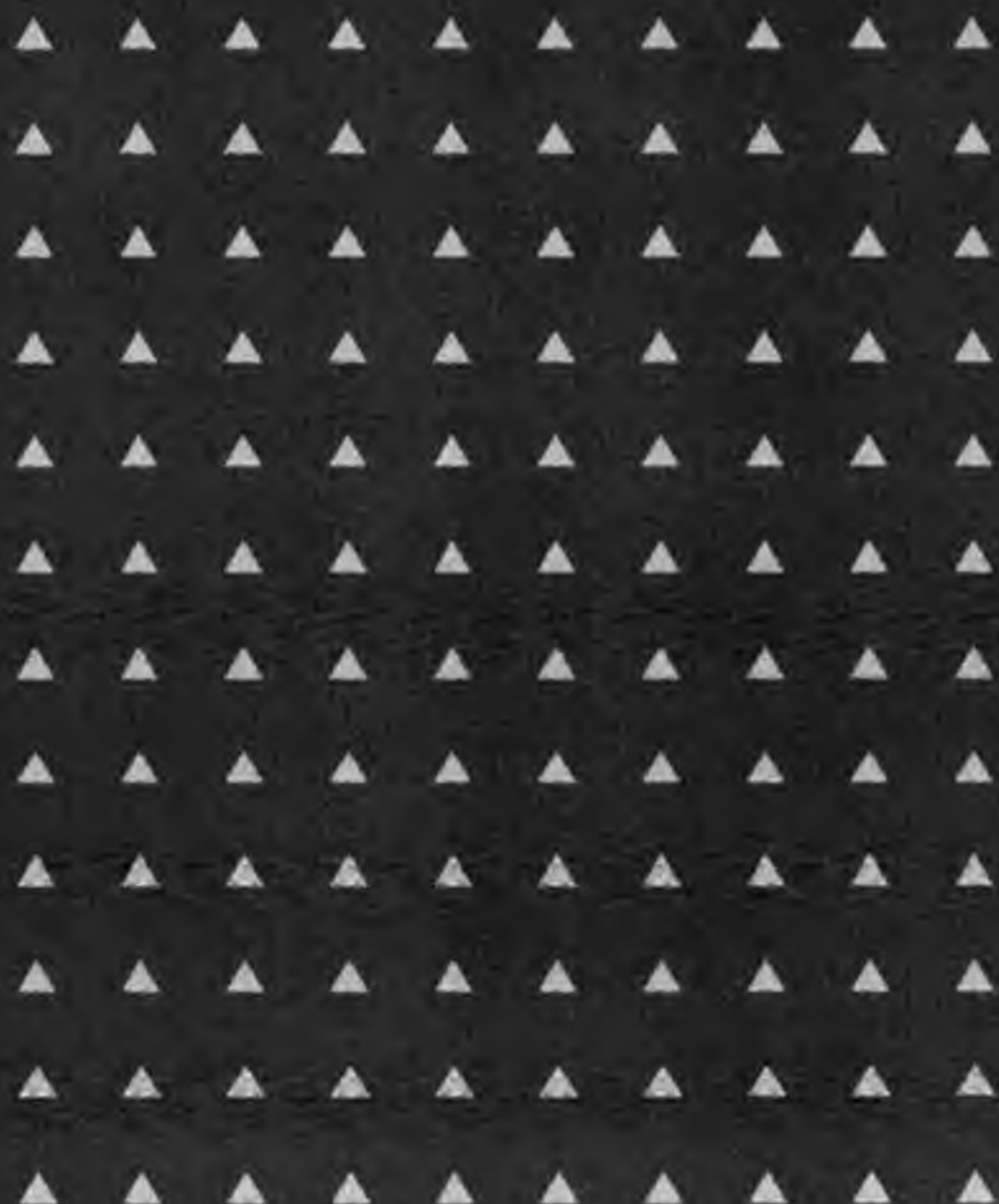
「あ、いや、もう結構です。やめて下さい。やめろってのにもう！ ジョイスティックぐらいはつけろ、大ボケッ」



# 第7章

## GAMMA5 ストラクチャーゲーム

- 7-1 ゲームの内容
- 7-2 プログラム入力方法
- 7-3 マップエディタ
- 7-4 敵攻撃データ
- 7-5 プログラムリスト





# 第7章 GAMMA5コンストラクションゲーム

## 7-1

### ゲームの内容

スクロールゲーム“GAMMA5”は、アクションゲームに、アドベンチャーやロールプレイングのおもしろさを取り入れた、512画面分に及ぶ大きさのゲームです。

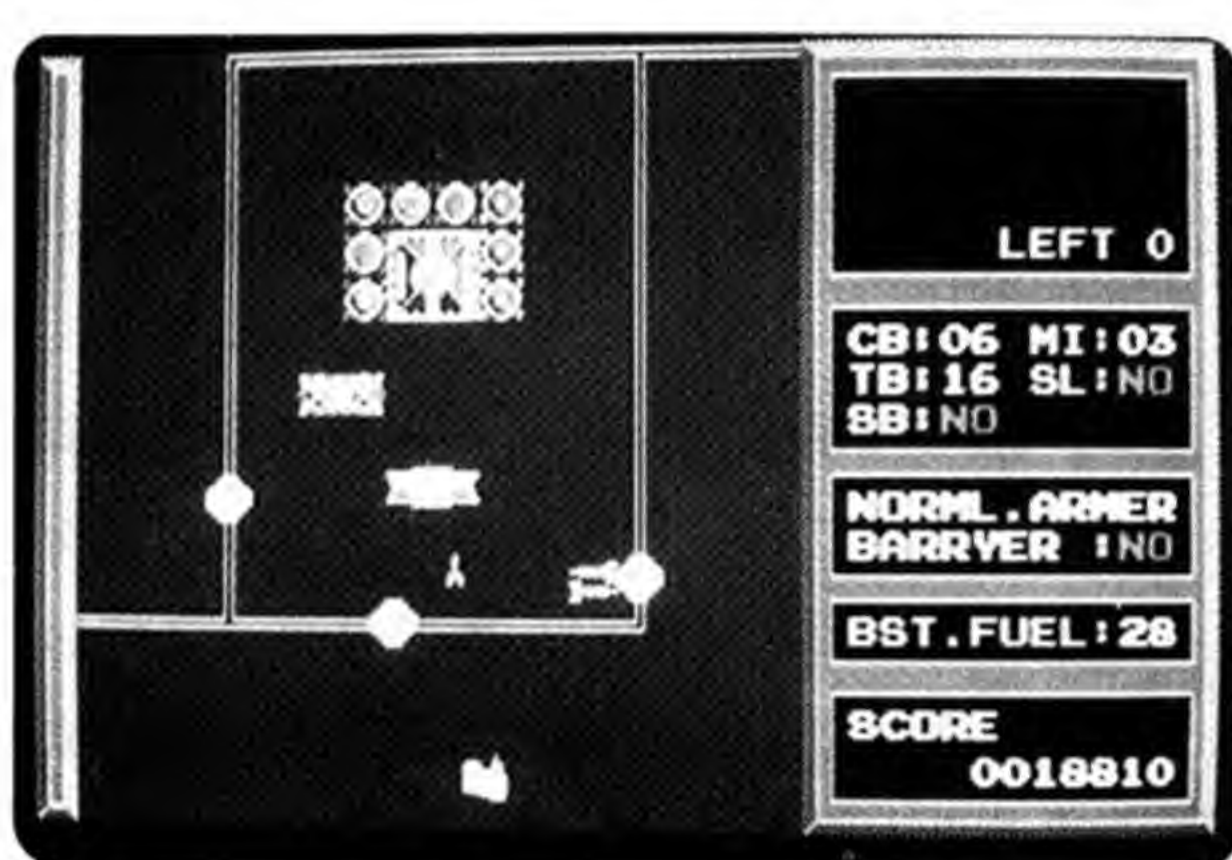
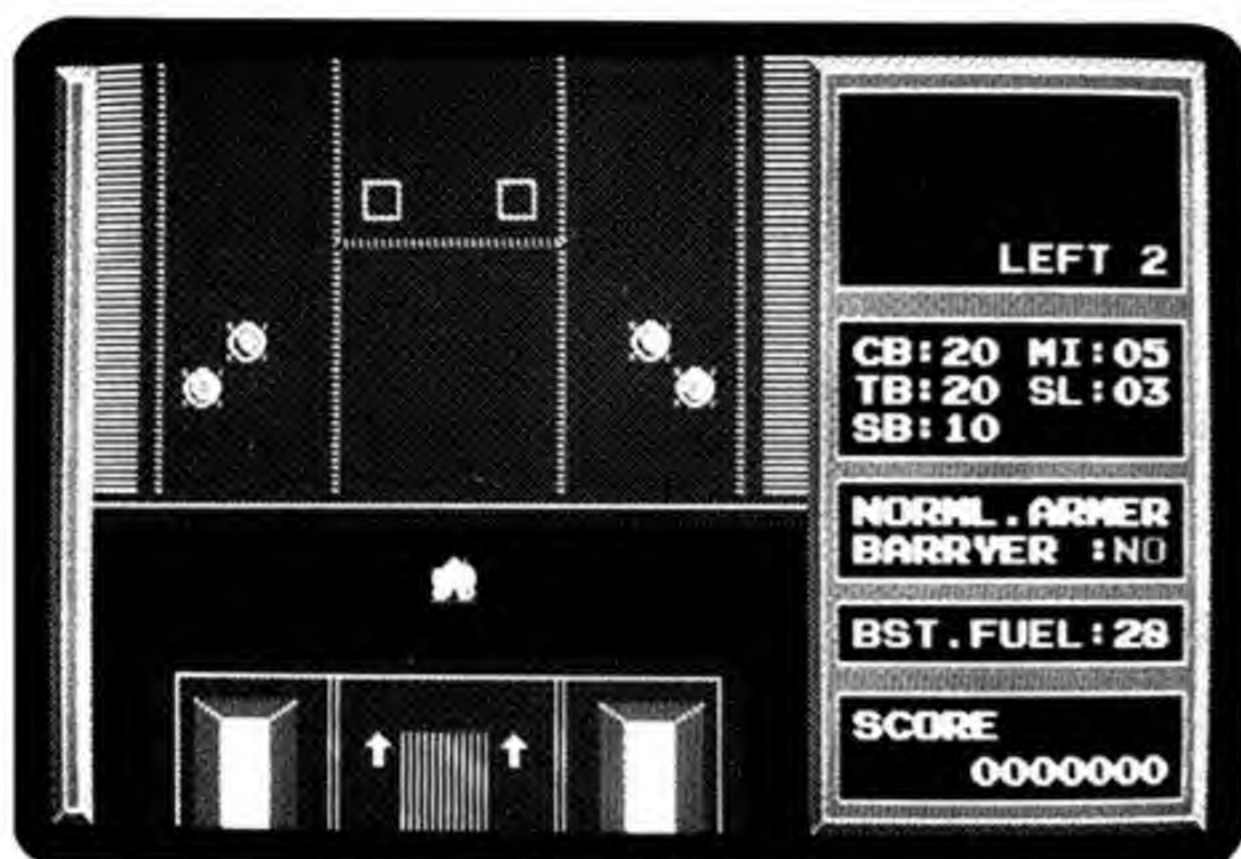
本章では、このゲームのアクション部分を取り出し、さらにコンストラクションを加えて、オリジナルスクロールゲームを楽しめるようにしました。

キャラクタの種類は製品の6分の1で、BGMありませんが、それでも256パターンの地表キャラクタに10種類の移動敵、6種の武器、様々な敵の攻撃などは製品と同じです。データにより簡単に難易度も変えられます。

### [登場キャラクタ]

- ① ウインドウ………ボールミサイル
- ② C. BOX………ボールミサイル
- ③ オッシレーター…アトミットウェーブ
- ④ ランチャー………パルスレーザー
- ⑤ フーマンチュー…ボールミサイル
- ⑥ ライトタワー……攻撃なし、反撃する
- ⑦ B. BOX………ボールミサイル、多数発射
- ⑧ バンデタワー……接近すると電撃を加える
- ⑨ タンク………反撃のみ
- ⑩ 広角砲………ボールミサイル
- ⑪ シューター………パルスレーザー





## ▲ GAMMA 5 名場面集

- ① V.ブロック……………ボールミサイル
- ② トーチカ……………ミサイル+パルスレーザー
- ③ モニター……………反撃のみ
- ④ ターミナル……………3種類それぞれある決った武器でしか破壊できない。破壊すると他の敵にもダメージを与える。くわしくは⑥
- ⑤ ゲート番……………レーザー以外の武器で倒すと、近くのゲートが開く
- ⑥ 隠れキャラクター……！や？の上を通ると出る。破壊するとアイテムになる



- ④バリアーA……………ミサイルで破壊可能
- ⑤バリアーB……………盾で体当たりすれば破壊可能
- ⑥クロック……………10発以上の弾で破壊可能。破壊後少しの間地表物の攻撃が止まる
- ⑦サイロ……………無防備
- ⑧ダブルシザーズ……………破壊不可能。はさまれるな!

## ●アイテム

- ①ツインビーム……………左右ななめ前方に発射する
- ②クロスビーム……………前後左右に同時発射する
- ③ソニックビーム……………壁などの障害物を透過する
- ④ミサイル……………格子ドアやバンデタワーなどを破壊できる
- ⑤ソーラーボール……………ほとんどのものを破壊できる。移動敵の中には撃たれたただけで逃げ出すものもある
- ⑥バリアーカートリッジ……………敵の弾を防ぐ
- ⑦バトルシールド……………体当たりで敵を倒せる
- ⑧スーパーシールド……………何度使っても壊れない
- ⑨ブースターフェル……………穴の上を飛んだり、倍速で移動する時に必要

## ●移動キャラクタ

- ①戦闘ロボット……………パルスレーザーを撃って来る
- ②使役ロボット(2種)……………追いかけて来る
- ③作業ロボット(2種)……………何もしない
- ④高速車両(2種)……………速いので衝突注意
- ⑤ガイドレール移動車両……………レールにそって追いかけて来る
- ⑥飛行物体(2種)……………フォーメーションで飛んで来るものと、壁に衝突して跳ねながら飛んで来るものとある

なお、敵の攻撃方法、ディフェンスの強さはデータにより変更可能ですが、あとの方で説明いたします。製品では、アイテムの多くは、敵を破壊したり、ある所を通過することによって出現するようになっていますが、本プログラムでは、シールドが必要ならば、あらかじめシールドをマップ上に置いておく必要があります。たとえば、穴の多い面を作ったならばブースターフェルがありませんと燃量切れてゲームオーバーという事になってしまいます。

## 7-2

### プログラム入力方法

プログラムが大変長いので、打ち込んでゲームを楽しむためには、ディスクが2枚必要です。

まず、DISK BASICの入ったディスクを一枚用意し、これをディスクAとします。

次に、フォーマット済みのディスクを用意し、これをディスクBとします。

いよいよ、リストの打ち込みですが、せっかく打ち込んでもミスがあつては苦勞がむくわれません。そこで、始めにチェックサムプログラムを走らせて各行ごとのチェックサムを確認しながら



ら作業しましょう。

リスト7-1を入力して下さい。変数Sはスタートのアドレスを、変数Eは終りのアドレスを示しています。打ち込むマシン語リストの終りのアドレスによって120行目のEの値を書き替えて下さい。

また、プリンタを使用されない方は、140行と、180行、230行は削除して下さい。

リスト7-2とリスト7-3はメインプログラムですから、必ずチェックサムの確認をして下さい。プログラムの入力が終わりましたら、次のようにしてディスクAにセーブします。

(i) リスト7-2

BSAVE "LIST1", &HC000, &H1F00

(ii) リスト7-3

BSAVE "LIST2", &HC000, &H1600

(iii) リスト7-4

BSAVE "DATA1", &HC000, &H1500

(iv) リスト7-5

BSAVE "DATA2", &HC000, &H2300

(v) リスト7-6

BSAVE "DATA3", &HC000, &H700

(vi) リスト7-7

BSAVE "DATA4", &HC000, &H2000

(vii) リスト7-8

BSAVE "DATA5", &HC000, &H1000

(viii) リスト7-9

BSAVE "DATA6", &HC000, &H400

これでほとんどのプログラム、データの作成が終わりました。もう一息です。

データの仕上げに、2Kバイトのランダムデータを作ります。

BASICで次のプログラムを書き、RUNして下さい。

10 CLEAR , &HBFFF

20 FOR I=&HC000 TO &HC7FF

30 POKE I, INT(RND(1)\*256)

40 NEXT

RUNさせてカーソルが戻ってきましたら、MONとしてEコマンドでC0000からC7FFまでが、00からFFまでのランダムデータで埋まっている事を確認して下さい。

このデータを次の様にセーブします。

BSAVE "RAND", &HC000, &H800

次にこのゲームの「世界」すなわち、マップを作るのですが、作り方は[7-3]マップエディタの所で説明してあります。そのマップエディタを利用して作ったデータか、または、リスト7-10、マップサンプルをモニタで打ち込み、

BSAVE "MAP1", &HC000, &H1000



としてセーブします。

最後にゲームをオートスタートさせるための、IPLプログラムを作ります。

リスト7-11 をモニタで打ち込み、BASICに戻して、

BSAVE "IPL", &HC000, &H100

としてディスクAに保存します。これは本プログラムが大変長いため、プログラムをロードするのに、DISK BASICが使えないためです。

では、ゲームディスク作りにかかりましょう。

ディスクAをドライブ1に、ディスクBをドライブ2にセットします。

念のために、ディスクAにはプロテクト用の銀のシールをノッチ(ディスクの切り込み)にはっておきます。

CLEAR, &HBFFF

とした後、次の手順でディスクBにマシン語を書き込みます。

(1) BLOAD "IPL"

で、"IPL" プログラムが &HC000 にロードされます。

MON

で、モニタにします。

h]^w2, 0, 0, 1, C000, C0FF

^wとはCTRL(コントロール)キーとWキーを同時に押す事を意味します。あとに続く数字の意味は、ディスクNO., サーフェースNO., トラックNO., セクタNO., 開始アドレス, 終了アドレスを表します。詳しくはマニュアルを参照して下さい。

^wでディスクに書き込む場合、書き込み場所を間違えるとディスクの内容を壊してしまう事がありますので、後に続く数字を絶対に間違わないように、落ち着いて入力して下さい。

ディスクにセーブできたかどうか確認するには、

h]^d2, 0, 0, 1

としますと、1セクタ分だけメモリダンプされます。うまくセーブできましたか?

次のステップからは、モニタに切り替える所は省略して、まとめて書いておきます。

(2) BLOAD "LIST1"

h]^w2, 0, 0, 2, C000, DEFF

(3) BLOAD "LIST2"

h]^w2, 0, 2, 1, C000, D5FF

(4) BLOAD "DATA1"

h]^w2, 0, 4, 2, C000, D4FF

(5) BLOAD "DATA2"

h]^w2, 0, 5, 7, C000, E2FF









(6) BLOAD "DATA3"

h]^w2, 0, 7, A, C000, C6FF

(7) BLOAD "DATA4"

h]^w2, 0, 9, 1, C000, DFFF



- (8) BLOAD "DATA5"   
 h]^w2, 0, B, 1, C000, CFFF 
- (9) BLOAD "MAP1"   
 h]^w2, 0, C, 1, C000, CFFF 
- (10) BLOAD "DATA6"   
 h]^w2, 0, D, 1, C000, C3FF 
- (11) BLOAD "RAND"   
 h]^w2, 0, D, 8, C000, C7FF 

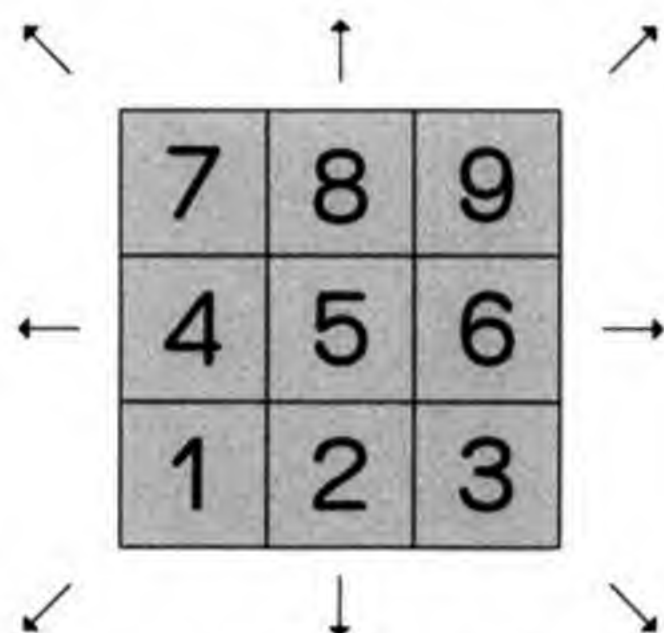
以上でゲームディスクの作成が終了しました。

ディスクBをドライブ1に入れて、リセットボタンを押して下さい。ロードにやや時間がかかりますが、ゲームがスタートするはずです(V1モードで実行)。

キーの操作方法を第7-1表にまとめておきます。ゲームオーバーになるまで一つの面を繰り返

第7-1表 "GAMMA 5" の操作方法

〔移動〕 テンキーにて行います。



\* 1, 3, 7, 9 のかわりに (4, 2), (2, 6), (4, 8), (6, 8) の二つのキーを同時に押すと、ななめに進めます。

\* ななめに進む時は向きが変わりません。また〔5〕のキーを押しながら、2, 4, 6, 8 の移動キーを押すと向きを変えずにその方向へ進めます。

〔攻撃〕 〔Z〕, 〔X〕, 〔C〕の三つのキーにて行います。またスペースキー、ファンクションキーを使用することもできます。

キ ー	名 称
〔SPACE〕 OR 〔C〕	レーザービーム
〔F.5〕 OR 〔X〕	クロスビーム
〔F.4〕 OR 〔Z〕	オーバーヘッドツインビーム
〔F.3〕 OR 〔X〕+〔C〕	ソニックビーム
〔F.2〕 OR 〔Z〕+〔X〕	ミサイル
〔F.1〕 OR 〔Z〕+〔X〕+〔C〕	ソーラーボール

〔スピードアップ〕 〔SHIFT〕か〔カナ〕キーを押すことにより倍速で動けますが、ブースターのエネルギーが減り、壁などに当たった時にペナルティが課せられます。

この他〔ESC〕キーで一旦停止し〔RETURN〕キーで再開します。



すわけですが、撃って来る弾数が増えるなど、だんだん難しくなって行きます。

## 7-3

### マップエディタ

サンプルマップはあくまでも一つの例であり、1MAP GAMEとして完成されたものではありませんし、前項で紹介しました移動キャラクタが全て登場するわけでもありません。

本プログラムにある、数々の機能を見ていただくために、ぜひマップデータや敵のデータを書き替えて、様々な面にチャレンジして下さい。

マップデータを作るには、一番簡単な方法はモニタを使ってデータを入力すればよいのです。

サンプルマップと、ゲーム画面を見くらべてみますと、データの並びと地形が似ているのがおわかりになると思います。

本プログラムはモニタを使用してマップデータが作りやすいように、横16のキャラクタで面を作り出しています。

しかし、やはりマップエディタを使って作成した方が、ずっとわかりやすいでしょう。さほど長いプログラムではないので、できるだけエディタを作ってみて下さい。

まず、マシン語プログラムを次のようにディスクAにセーブします。

#### (i) リスト7-12

BSAVE "EDIT", &HC000, &H500

#### (ii) リスト7-13

CLEAR, &HBDFE

としてから入力して下さい。

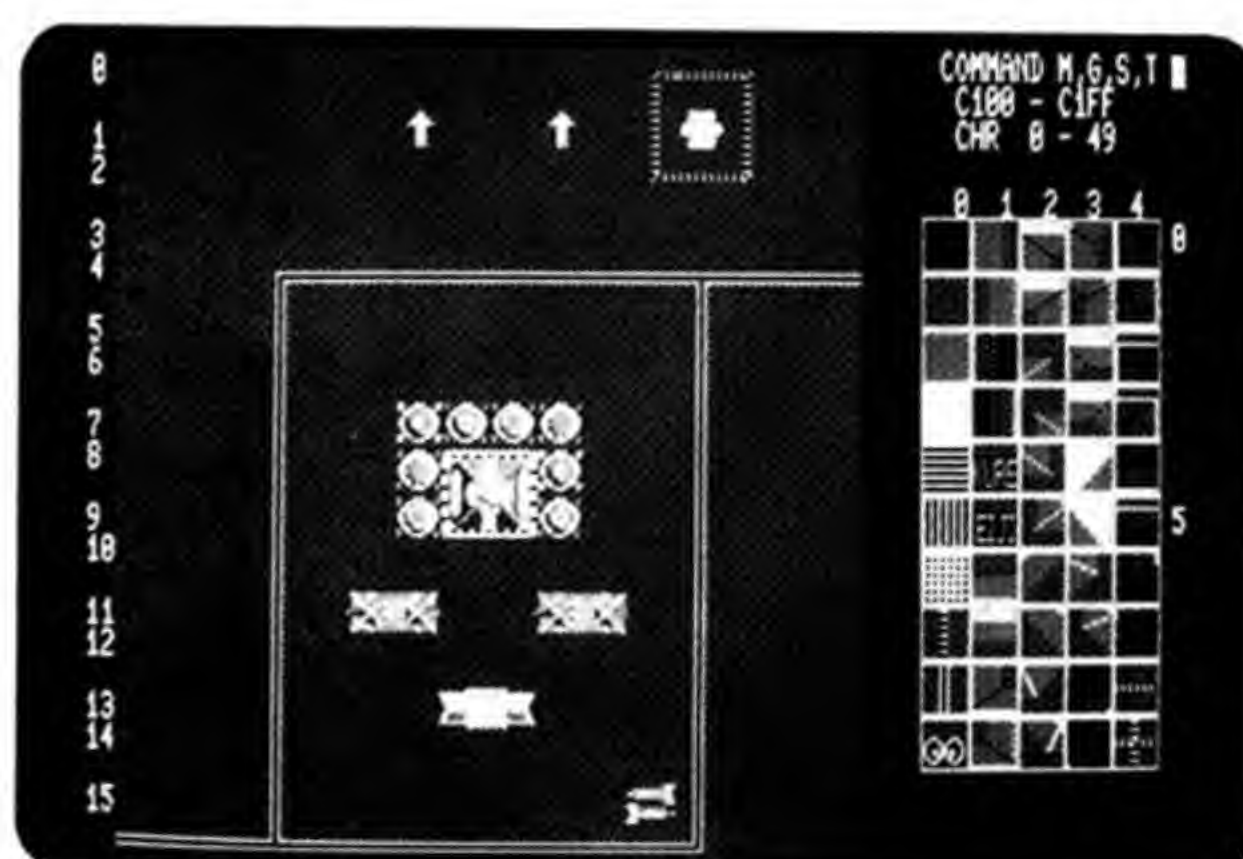
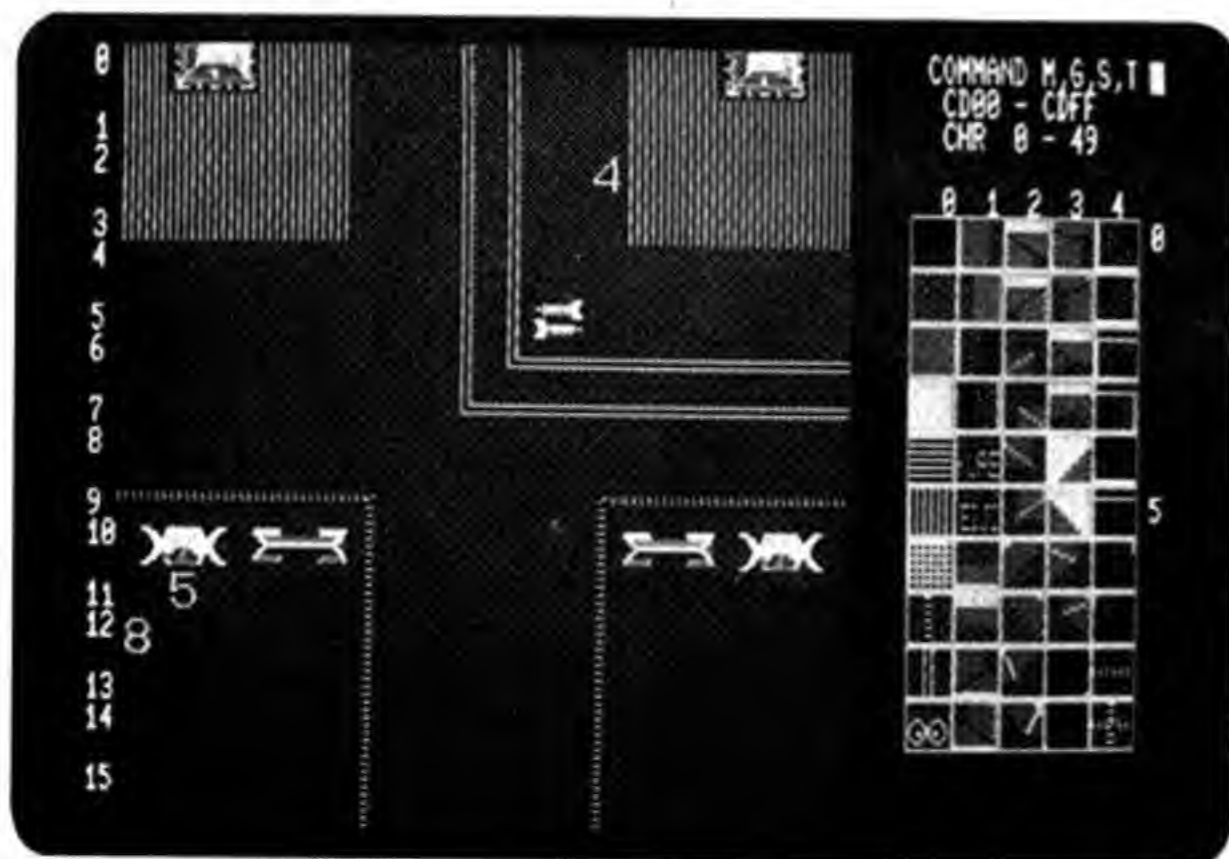
BSAVE "GRGETT", &HBE00, &H200

#### (iii) リスト7-14

BSAVE "ZSCENT", &HD100, &H100

BASICプログラムからロードするリストもありますので、ファイル名も間違いなくセーブして下さい。

次にBASICプログラムを入力します。



▲オリジナルマップを作ろう！



## (iv) リスト7-15


SAVE "MAPED" 

以上でマップエディタが完成しました。

では、プログラムのロード方法です。

① BLOAD "EDIT", &H3800 

② BLOAD "DATA1", &H4100 

③ BLOAD "DATA2", &H5600 

④ BLOAD "DATA3", &H7900 

②, ③, ④は前節で入力したデータです。ロードアドレスの指定をつけますが、确实に行って下さい。

以上のプログラムは、一度ロードすればBASICプログラムをストップしても、再度ロードする必要はありません。

ここで、編集済みのマップや、サンプルマップから作り直す場合には、マップデータをロードしておきます。

(例) BLOAD "MAP1" 

マップデータを読まずにマップを作る時には、メモリを初期化しておくといいでしょう。

MON 

h] FC000, D00F, 1 

として、マップのメモリ領域をコード01で埋めておきます。

次にBASICプログラムをロードして走らせます。

RUN "MAPED" 

と入力しますと、C000番地からのマップが表示されます。もし、メモリを01で埋めているならば、スクリーンの左半分はグリーンのアミで埋まっているはずです。このグリーンのアミが戦士の進める「床」になります。

マップエディタは、四つの基本的な機能があります。画面、右上に、"COMMAND M, G, S, T" と表示され、カーソルが点滅してコマンド待ちになっていますね？ この時、M, G, S, Tのどれかを押すと次の作業を行います。

●Mコマンド=編集画面を表示する。

[M] キーを押すと、"(C000—)" と表示され、入力待ちになります。表示するメモリのアドレスを聞いているのです。C000～CF00の範囲で16進数を入力します。最後の1ケタは無視されます。

●Gコマンド=セットするキャラクタを呼び出す。

[G] キーを押すと、"TOP CHR NO." と表示され、入力待ちになります。画面右に表示されているキャラクタの数は50個ですが、マップにセットできるのは256種あります。

エディタでセットできるキャラクタは、右の50個の中になければなりません。そこで、先頭のキャラクタ番号を入力して、表示するキャラクタを変えるわけです。

入力できる数字は0から210まで、10進数で入力して下さい。



## ●S コマンド=キャラクタをセットする。

[S] キーを押すと、右の50個のキャラクタのうち、一つが点滅する白いワクで囲まれます。このワクをテンキーの2, 4, 6, 8で上下左右に動かし、マップ上にセットしたいキャラクタの所へもって行きます。

そこで、5のキーを押しますと、左のマップ中程にアスタリスク(\*)が点滅します。このマークをまたテンキーの2, 4, 6, 8を使って、セットしたい位置に移動させ、5のキーを押すと選んだキャラクタがプットされるはずです。

アスタリスクの上下移動は変な間隔で動きますが、大体望み通りの所にプットできると思いま

5のキーを押す度に、選んだキャラクタが置かれて行きますが、他のキャラクタに変えたい時は、スペースバーを押して下さい。再び右のキャラクタ中の白ワクが点滅し、選択モードとなります。

画面にないキャラクタを呼び出したり、他のモードにしたい時は、リターンキーを押して下さい。コマンドの選択に戻ります。

## ●T コマンド=スクロールテスト

[T] キーを押すと、"TEST OK? (Y/N)"と聞いて来ます。[Y]を押すとC000からCFFFまでのマップをスクロールさせてみます。キャラクタのセットのし方が悪いと、キャラクタのわだちが残り、書きたくない所まで書いてしまったりします。


スクロールテストを行いたくない時は[N]を押して下さい。

マップを作り終えたら、自分の好きなファイル名でディスクにセーブしておきます。

まず[STOP]キーでエディタを中断させ、


CLS 3 

で画面をクリアしてから、

BSAVE "ファイル名", &HC000, &H1000 


で、セーブできます。

この作ったマップで遊ぶ時は、

BLOAD "ファイル名" 

でロードしたのち、ディスクBをドライブ2にセットして、

MON 

h]^w2, 0, C, 1, C000, CFFF 

とします。念のため、ディスクBのバックアップを作っておくとよいでしょう。

では、次にマップで使われるキャラクタを番号順に表にまとめたものがありますので、ご覧下さい(第7-2表)。

表では240種類のキャラクタまでの説明があり、それ以降は書いてありません。

ところが、たとえばサンプルマップのCDB1番地を見ますと、"F5"が書かれています。戦士がこの上を通過すると五つ前方に隠れキャラクタが出現し、これを破壊すると、エリア5へ入る落とし穴が現れますが、これは市販ゲームでの話です。このように、表に説明のないもの、意味なしと書かれているもの同様、240番以降のコードはアドベンチャーゲーム用のシークレットコードに使われているので、本プログラムでは使用しません。



第7-2表 キャラクターデータ紹介

No.	HEXNo.	名 称	機 能
0	0 0	穴	ブースターで飛ぶ
1	0 1	床	通常の通路
2	0 2		
3	0 3	天井	
4	0 4		
5	0 5		
6	0 6		
7	0 7	ライン	
8	0 8	ガイドレール	ガイドレール車両の道筋用
9	0 9	目玉	ムービングバリア用
10	0 A	) 壁	
11	0 B		
12	0 C		
13	0 D	) 穴のふち	
14	0 E		
15	0 F		
16	1 0	} 壁	
17	1 1		
18	1 2		
19	1 3		
20	1 4		
21	1 5		
22	1 6		
23	1 7		壁は障害物となりますから、無理のないように配置します
24	1 8		またキャラクタ同志がうまくつながるように、スクロールして
25	1 9		おかしくならないように、確かめてセットして下さい
26	1 A		
27	1 B		
28	1 C		
29	1 D		
30	1 E		
31	1 F		
32	2 0	} 穴のふち	
33	2 1		
34	2 2		
35	2 3		
36	2 4		
37	2 5		
38	2 6		
39	2 7		
40	2 8		
41	2 9		穴を多くする時は、フエル(キャラクタNo. 232)を適当に配置する
42	2 A		ようにした方がよいです
43	2 B		
44	2 C		
45	2 D		
46	2 E		
47	2 F		
48	3 0	} ライン	
49	3 1		
50	3 2		
51	3 3		
52	3 4		
53	3 5		
54	3 6		
55	3 7		
56	3 8		
57	3 9		
58	3 A	} ビックリマーク	
59	3 B		五つ前方に隠れキャラが出る



No.	HEXNo.	名 称	機 能
60	3 C	ワク	B. B O Xが出現する
61	3 D	落とし穴	) 本プログラムでは意味なし
62	3 E	丸ワク	
63	3 F	落とし穴	
64	4 0		
65	4 1	} ガイドレール	ガイドレール 移動車両用
66	4 2		
67	4 3		
68	4 4		
69	4 5		
70	4 6		
71	4 7		
72	4 8		
73	4 9	} ハテナマーク	五つ前方に隠れキャラが出現する
74	4 A		
75	4 B		
76	4 C		
77	4 D	} 矢印	
78	4 E		
79	4 F		
80	5 0		
81	5 1	} 格子ドア	ミサイル、シールドで破壊できる
82	5 2		
83	5 3		
84	5 4		
85	5 5		
86	5 6		
88	5 7		
87	5 8	} シャッター	門番が近くにいた時は、門番を破壊するとオープンする *80~95までのドアは必ず2個組みにして、正常な形で置いて下さい
89	5 9		
90	5 A		
91	5 B		
92	5 C		
93	5 D		
94	5 E		
95	5 F		
96	6 0	} ドア	常に通過可能
97	6 1		
98	6 2		
99	6 3		
100	6 4		
101	6 5		
102	6 6		
103	6 7		
104	6 8	} 模様	模様の終りと先頭に書く途中は2, 4, 5のキャラクタを使用
105	6 9		
106	6 A		
107	6 B		
108	6 C		
109	6 D		
110	6 E		
111	6 F		
112	7 0	} 通常使用せず	
113	7 1		
114	7 2		
115	7 3		
116	7 4		
117	7 5		
118	7 6		
119	7 7		
		番号キャラクタ	市販のゲームでは暗号として使用されている所もある 本プログラムでは意味なし



No.	HEXNo.	名 称	機 能
120	7 8	模様	4 個組みで使用
121	7 9		
122	7 A		
123	7 B		
124	7 C		
125	7 D	上へ昇る階段 下へ降る階段	) 本プログラムでは意味なし
126	7 E		
127	7 F	壁	
128	8 0		
129	8 1		
130	8 2		
131	8 3		
132	8 4		
133	8 5		
134	8 6		
135	8 7		
136	8 8		
137	8 9	キャラクタ破壊跡	} マップ作りにはあまり使用しません
138	8 A		
139	8 B		
140	8 C		
141	8 D		
142	8 E	ウインドウ左	以下のキャラクタ機能は本文を参照して下さい
143	8 F		
144	9 0		
145	9 1		
146	9 2		
147	9 3	ウインドウ右	
148	9 4		
149	9 5		
150	9 6		
151	9 7		
152	9 8	C. BOX	
153	9 9		
154	9 A		
155	9 B		
156	9 C		
157	9 D	オッシレーター	
158	9 E		
159	9 F		
160	A 0		
161	A 1		
162	A 2	ランチャー	
163	A 3		
164	A 4		
165	A 5		
166	A 6		
167	A 7	フーマンチュー	
168	A 8		
169	A 9		
170	A A		
171	A B		
172	A C	ライトタワー	
173	A D		
174	A E		
175	A F		
176	B 0		
177	B 1	B. BOX	
178	B 2		
179	B 3		
		バンデタワー	
		タンク	
		広角砲	
		意味なし	



No.	HEXNo.	名 称	機 能
180	B 4	) シューター	* これ以後の組み合わせキャラクタは必ず正常な形で左右、あるいは前後左右組み合わせて配置して下さい
181	B 5		
182	B 6	) V. ブロック	
183	B 7		
184	B 8	) トーチカ	
185	B 9		
186	B A		
187	B B	) モニタ	
188	B C		
189	B D		
190	B E	) 破壊跡	
191	B F		
192	C 0		
193	C 1	) 隠れキャラクタ	
194	C 2		
195	C 3		
196	C 4	) 味方	
197	C 5		
198	C 6		
199	C 7	) ダブルシザーズ	
200	C 8		
201	C 9		
202	C A	) ターミナル	
203	C B		
204	C C		
205	C D	) ゲート番	
206	C E		
207	C F		
208	D 0	) バリアー A	
209	D 1		
210	D 2		
211	D 3	) バリアー B	
212	D 4		
213	D 5		
214	D 6	) クロック	
215	D 7		
216	D 8		
217	D 9	) サイロ	
218	D A		
219	D B		
220	D C	) ツインビーム	
221	D D		
222	D E		
223	D F	) クロスビーム	
224	E 0		
225	E 1		
226	E 2	) ソニックビーム	
227	E 3		
228	E 4		
229	E 5	) ミサイル	
230	E 6		
231	E 7		
232	E 8	) ソーラーボール	
233	E 9		
234	E A		
235	E B	) バリアーカートリッジ	
236	E C		
237	E D		
238	E E	) バトルシールド	
239	E F		



マップエディタで表示されるキャラクタを見ますと、同じキャラクタが別の番号で2個出て来る事に気づかれると思いますが、これは大変重要な意味を持ちます。

例として、縦ラインのキャラクタが、7と50の両方に入っていますね。ある場面に縦のラインを引こうとして、7を並べてスクロールさせます。

するとどうでしょう。縦のラインは出現しません！逆に、50のキャラクタを並べてスクロールさせます。

すると、ラインは現れますが、場合によってはスクロールが遅くなる事があります。

第6章で書きましたが、「GAMMA5」はキャラクタスクロールタイプのスクロールゲームです。すべてのキャラクタをスクロールさせるわけではありません。いくつかのキャラクタをスクロールさせ、他の部分は変化しないか、またはキャラクタのスクロールした「跡」として作られます。

つまり、縦のラインを作ろうとする時、50番のキャラクタただ一つを書けば、スクロールさせると、その後になてのラインが残ってゆくのです。

このように、15番までのキャラクタはスタート時のみ画面に書かれ、スクロールしている時は画面に書かれません。だから、16番以上のキャラクタが多いと、スクロールが遅くなる事があるわけです。

エディットする一画面は256個のキャラクタで作られますが、なるべく16番以上のキャラクタは50個以内にしてください。

もう一つ注意していただきたいのは、NO. 9、目玉のキャラクタの配置についてですが、これはエディタではテストできません。ゲームをしてみると、キャラクタコード9の上にムービングバリアが登場して、9の列の端から端までを往復します。

ムービングバリアは、画面に4個までしか登場できず、横一ラインに2個出現させる事はできません。また、一列に並んだコード9の右端は必ずコード1を置いて、画面の右端までムービングバリアを動かす時には二つ分コード1を置いて下さい。

以上、キャラクタの配置にはいろいろと制限があって複雑のようですが、実際にやってみてスクロール画面が乱れたり、ムービングブロックが画面を飛び出したりしないようにすればよいのです。

キャラクタの置き方によっては一見床のように見える穴や、通過できる壁ができます。

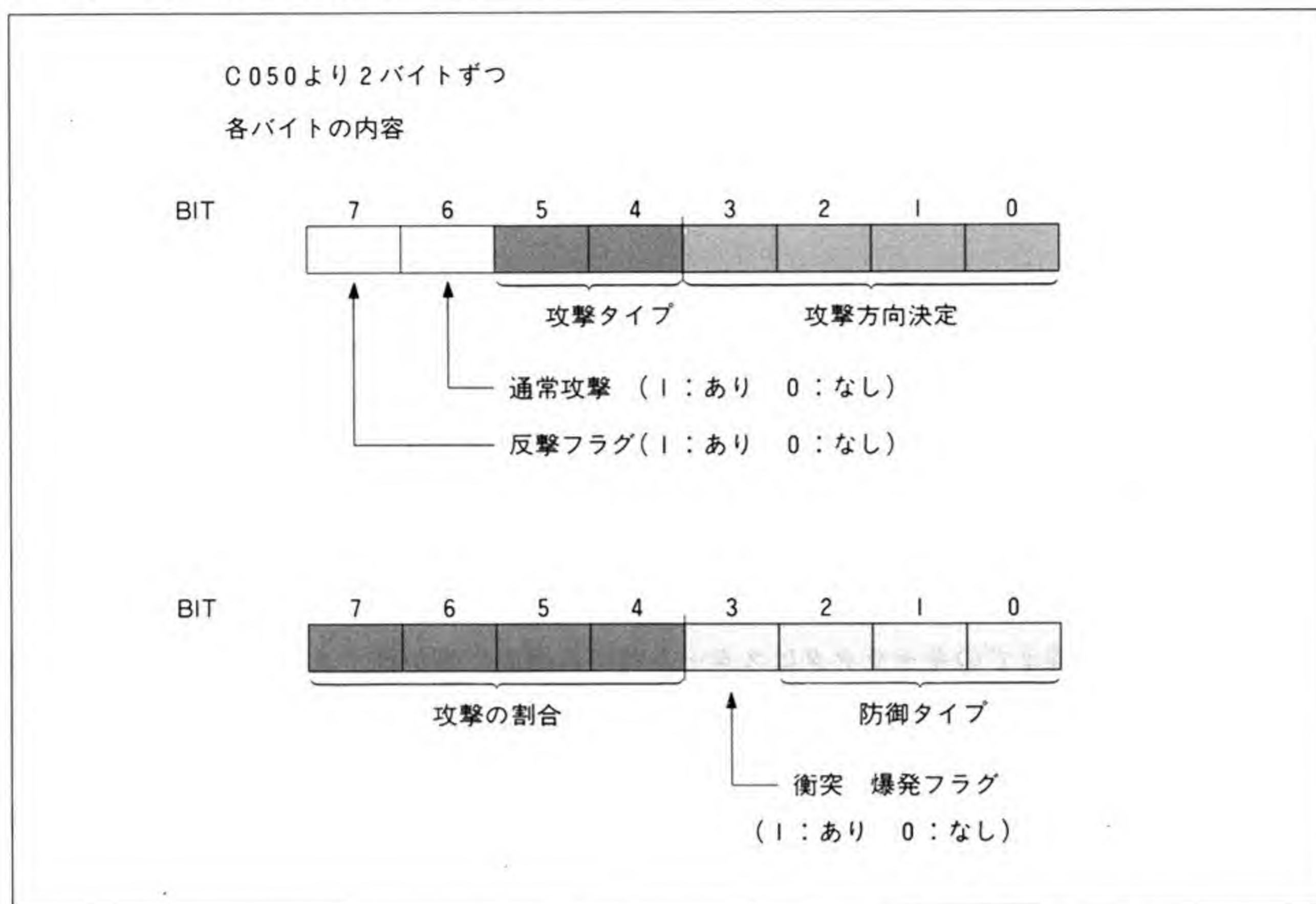
## 7-4 敵攻撃データ

リスト7-9、「DATA6」のC000からC0FFまでを見て下さい。ここには敵の攻撃をコントロールするデータが入っていて、この部分を書きかえる事により、攻撃のし方、激しさ、敵の出現位置、敵の種類、強さなどすべてがコントロールできます。

まず、地上物の攻撃データは、キャラクタNO. 144から191までの一つ一つに関して、攻撃、防御を決めるのが、C050からC0EFまでに収められています。キャラクタ一つに対してデータは



第7-3表 地上キャラクタの攻撃・防御データ



二つ、始めの方が攻撃に関するもの、後の方が半分は攻撃率、半分が防御に関するものです（第7-3表）。

このうち、攻撃タイプと防御タイプについて説明します。

#### ●攻撃タイプ

- 0：攻撃なし
- 1：ボールミサイル
- 2：パルスレーザー
- 3：アトミックウェーブか、または、ボールミサイル多数発射、オッシレーターのみアトミックウェーブを発射する。

#### ●防御タイプ

- |                                     |               |
|-------------------------------------|---------------|
| 0：防御なし。どんな武器でも倒せる                   | 4：ツインビームのみ有効  |
| 1：レーザー無効、他の武器で倒せる                   | 5：クロスビームのみ有効  |
| 2：全ビーム無効                            | 6：ソニックビームのみ有効 |
| 3：全武器無効。衝突爆発フラグが1ならば、<br>シールドで破壊できる | 7：ソーラーボールのみ有効 |

このように一つ一つのキャラクタが特有の攻撃、防御を持つため、四つのパターンに変化するキャラクタでは、状態によって設定を変えられるわけです。たとえば、形により発射方向を変えたり、口を開いた時に攻撃して、口を閉じた時は防御力が大きいなどです。

地上物のデータは少し難しいですが、全体の攻撃割合を変えるのは簡単です。



リスト7-9のC025を見て下さい。03が入っていますね。この数字を大きくすると攻撃が激しくなり、少なくすると攻撃が減ります。

一度スクロールが最後まで行くと、この数字は自動的に大きくなりますから、あまり大きくはしない方がよいでしょう。

次に、移動して攻撃してくる敵のデータを説明します。マップによってこのデータは必ず書き替えなければなりません。

エディタによってマップを作ったら、マップのどこの所で敵を出現させたいのか決めます。Mコマンドで出現させたいシーンを見つけ、その時に入力したアドレスを書きとめておきましょう。

このアドレスを指定する場所がC000からC013までの20バイトで、2バイトずつ、10個まで指定できます。始めの2バイト、すなわち、C000番地とC001番地を見てみますと、“40CE”と入っています。これは、CE40番地からの画面が表示された時に、敵が出現する事を現わします。Z80では、2バイトの数は上下さかさまになるのでしたね。

指定する番地の最後は0でなければなりません。つまり、CE10と指定すれば問題はありませんが、CE12と指定したのでは敵は出現しません。

さて、次に指定した所にどんな敵を出現させるか考えましょう。

敵のタイプを決める数字は、C0F0からC0F9までの間に入っています。

リスト7-9では、CE40でコード05の敵が、CD00ではコード04の敵が出現するようになっています。このコードの上のケタ(上の例では0)は、移動して来る敵の能力を決める数で、下のケタが敵のタイプを決めます。

敵のタイプとコードの下のケタは次のように対応します。

コード 1	…	ガイドレール移動車両
コード 2, 3	…	高速車両
コード 4～6	…	作業ロボット
コード 7～9	…	使役ロボット
コード A, B	…	戦闘ロボット
コード C, D	…	フォーメーション飛行物体
コード E, F	…	飛行物体(壁に当たるとはねかえる)

各種類についてコードが複数割り当ててあるのは、キャラクターが異なったり、出現のし方が一度に全部出るか一つずつ出るかなどの違いがあるからです。

百聞は一見にしかずで、実際に数字をいろいろと置き替えてみますと、今まで出て来なかったキャラクターも登場するでしょう。

そのうちに、その場面にふさわしい敵が出現できるようになると思います。

コード01のガイドレール移動車両は、画面の端にガイドレールが来ていなければ出現できませんし、フォーメーションで動くものは、壁があってもつき抜けて来ます。場面を考える必要がありますね。

コードの下のケタの数は以上でおわかりいただけたかと思いますが、上のケタはまだ謎のままです。

では、ガイドレール移動車両の場合を例にして、上のケタの数が変わるとどうなるかみてみます。




- 01 ... レーザーで破壊でき、レールの上を追いかけて来る普通の車両
- 11 ... レーザーでは破壊できない
- 21 ... 弾を撃って来るいやな奴
- 31 ... レーザーで破壊できないし、弾も撃って来るスーパー車両!

この様に、上のケタには0, 1, 2, 3のどれかを入れて使います。ロボットでは4も使う事があるのですが、どうなるかは、やってみてのお楽しみ。ただし、上のケタには8以上の数字は入れないで下さい。出現と同時に自爆してしまうでしょう。

さて、この様にして、敵のデータを作り上げましたら、さっそく、ゲームディスクに書き込んで、オリジナルゲームをやってみましょう。

ディスクBをドライブ1にセットします。

h)^ w1, 0, D, 1, C000, C0FF 

とモニタから書き込みます。

リセットボタンを押すと、あなただけの「GAMMA 5」がスタートするでしょう。

#### リスト7-1 チェックサムプログラム

```

100 CLEAR,&HBFFF:DIM D(16,16):C=0
110 S=&HC000
120 E=&HDEF0
130 IF C=0 THEN GOSUB 470
140 PRINT USING "& &: ";HEX$(S);
150 LPRINT USING "& &: ";HEX$(S);
160 FOR J=1 TO 16:B=PEEK(S):D(C,J)=B
170 B$=RIGHT$("0"+HEX$(B),2)
180 PRINT USING "&& ";B$;
190 LPRINT USING "&& ";B$;
200 IF J=8 THEN PRINT " ";LPRINT " ";
210 U=U+PEEK(S) AND &HFF:S=S+1
220 NEXT J:D(C,0)=U
230 U$=RIGHT$("0"+HEX$(U),2)
240 PRINT " ";U$:U=0
250 IF C=15 THEN LPRINT CHR$(27);"B";
260 LPRINT " ";U$:U=0
270 C=C+1:IF C=16 THEN GOSUB 300
280 IF S<=E THEN 130
290 END
300 FOR J=0 TO 16:D(16,J)=0:NEXT
310 FOR J=0 TO 16:FOR I=0 TO C-1
320 D(16,J)=D(16,J)+D(I,J):NEXT I:NEXT J:C=0
330 PRINT "-----"
340 LPRINT "-----"
350 PRINT "sum : ";
360 LPRINT "sum : ";
370 FOR J=1 TO 16
380 B$=RIGHT$("0"+HEX$(D(16,J)),2)
390 PRINT USING "&& ";B$;
400 LPRINT USING "&& ";B$;
410 IF J=8 THEN PRINT " ";LPRINT " ";
420 NEXT J
430 U$=RIGHT$("0"+HEX$(D(16,0)),2)
440 PRINT " ";U$:PRINT
450 LPRINT CHR$(27);"A";LPRINT " ";U$:LPRINT
460 RETURN
470 PRINT "      +0 +1 +2 +3 +4 +5 +6 +7  +8 +9 +A +B +C +D +E +F :sum"
480 LPRINT "      +0 +1 +2 +3 +4 +5 +6 +7  +8 +9 +A +B +C +D +E +F :sum"
490 RETURN

```



## リスト7-2 GAMMA5 メインプログラム1 "LIST1"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: 31 00 40 C3 75 24 3E 06 D3 44 3E 18 D3 45 3E 08 :DC
C010: 03 44 3E 10 D3 45 3E 08 D3 44 3E 50 D3 45 3E 0C :CD
C020: 03 44 3E 18 D3 45 3E 0D D3 44 3E 01 D3 45 3E 07 :83
C030: 03 44 3E 07 D3 45 3E 03 06 D3 44 3E 50 D3 45 3E :7C
C040: 08 D3 44 3E 10 D3 45 3E 08 D3 44 3E 10 D3 45 3E :89
C050: 0C D3 44 3E 03 D3 45 3E 0D D3 44 3E 0F D3 45 3E :81
C060: 07 D3 44 3E 07 D3 45 3E 3E 02 D3 44 3E 0D D3 45 :CE
C070: 3E 03 D3 44 3E 01 D3 45 3E 09 D3 44 3E 10 D3 45 :73
C080: 3E 08 D3 44 3E 20 D3 45 3E 0C D3 44 3E 18 D3 45 :A5
C090: 3E 0D D3 44 3E 01 D3 45 3E 07 D3 44 3E 3D D3 45 :A8
C0A0: C9 3E 02 D3 44 3E 77 D3 45 3E 03 D3 44 3E 00 D3 :56
C0B0: 45 3E 09 D3 44 3E 10 D3 45 3E 08 D3 44 3E 50 D3 :CA
C0C0: 45 3E 0C D3 44 3E 18 D3 45 3E 0D D3 44 3E 01 D3 :88
C0D0: 45 3E 07 D3 44 3E 3D D3 45 3E 0C D3 44 3E 0E 5A :E1
C0E0: ED 79 0C 10 F8 C1 C9 C5 06 08 0E 5A AF ED 79 3C :80
C0F0: 0C 10 FA C1 C9 C0 E7 01 3E 07 D3 57 C9 CD F5 01 :50
sum : 10 E1 63 95 96 14 57 82 E7 F5 93 5D 2C 0C E8 4E :A6

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C100: 3E 01 D3 59 C9 CD F5 01 3E 06 D3 56 3E 02 D3 58 :CF
C110: 3E 01 D3 59 C9 CD F5 01 3E 05 D3 58 3E 01 D3 59 :D0
C120: C9 CD F5 01 3E 01 D3 58 3E 01 D3 59 C9 C5 3E 20 :4D
C130: D3 40 3E 06 D3 54 0E 14 06 FF 10 FE 0D 20 F9 AF :88
C140: D3 54 D3 40 C1 C9 C5 0E 80 3E 20 D3 40 06 80 10 :1E
C150: FE AF D3 40 06 80 10 FE 0C 20 EE C1 C9 AF C5 E5 :51
C160: D3 5C 77 23 10 FC E1 C1 C5 E5 D3 5D 77 23 10 :F7
C170: E1 C1 C5 E5 D3 5E 77 23 10 FC E1 C1 CD 80 06 0D :25
C180: 20 DC D3 5F C9 06 02 0E 04 18 1C 06 03 0E 0C 18 :80
C190: 16 06 01 0E 08 18 10 06 02 0E 04 18 0A 06 0E 0E :B1
C1A0: 0C 18 04 06 06 0E 18 CD CE 02 D8 CD B6 02 CD 80 :A1
C1B0: 06 D8 0D 20 F6 C9 C5 E5 AF D3 5C 77 23 10 FC E1 :D9
C1C0: C1 C5 E5 D3 5D 77 23 10 FC E1 C1 D3 5F C9 E5 D5 :98
C1D0: 11 00 C0 A7 ED 52 D1 E1 C9 0E 08 18 02 0E 04 CD :41
C1E0: CE 02 D8 AF D3 5C 77 D3 5D 77 D3 5F CD 80 06 08 :01
C1F0: 0D 20 F1 C9 06 03 0E 0C AF C5 E5 D3 5D 77 23 10 :3D
sum : 92 E8 0E C6 3D AF 60 F4 75 70 20 36 10 34 25 8F :C1

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C200: FC D3 5F E1 C1 CD 80 06 D8 0D 20 ED C9 0E 08 AF :A3
C210: D3 5D 77 D3 5F CD 80 06 D8 0D 20 F4 C9 CD 87 06 :48
C220: AF D3 5D 77 23 77 D3 5F C9 E5 C5 D3 5C 72 23 10 :69
C230: FC C1 E1 0D 28 13 CD 80 06 E5 C5 73 23 10 FC C1 :46
C240: E1 0D 28 05 CD 80 06 18 E0 D3 5F C9 11 6F 03 18 :FC
C250: 0D 11 73 03 18 08 11 77 03 18 03 11 7B 03 C5 06 :84
C260: 04 D3 5C 1A 13 77 CD 80 06 10 F8 D3 5F C1 C9 EE :DC
C270: 3B 8E 23 ED 87 DE FB 89 26 98 62 EE 3B 8E 23 E5 :D1
C280: C5 CD 91 03 C1 E1 CD 80 06 23 05 05 0D 0D C3 5D :82
C290: 02 05 05 CD A1 03 CD B1 03 05 11 50 0D A7 ED 52 :1A
C2A0: D1 C5 E5 23 3E FF D3 5C 77 23 10 FC D3 5F E1 C1 :84
C2B0: C9 D5 D3 5C 16 01 1E 80 C5 E5 7E B2 77 23 23 10 :29
C2C0: FD 7E B3 77 E1 CD 80 06 C1 0D 20 EC D3 5F D1 C9 :7F
C2D0: D3 5C C5 D5 E5 06 07 1A 77 23 13 1A 77 2B 13 CD :1E
C2E0: 80 06 10 F3 D3 5F E1 D1 C1 C9 D3 5D 18 E4 D3 5E :54
C2F0: 18 E0 06 08 3E 60 D3 5D 77 CD 80 06 10 FA D3 5F :DA
sum : 70 6F 0A DD A7 77 45 DE 43 3D B0 2E 00 BC A0 4A :0B

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C300: C9 CD 87 06 3E FF D3 5D 77 23 77 D3 5F C9 11 28 :D5
C310: 04 18 03 11 2C 04 06 04 1A D3 5C 77 D3 5D 77 CD :9E
C320: 80 06 13 10 F3 D3 5F C9 C0 30 0C 03 D3 0C 30 C0 :95
C330: E5 CD 3B 04 E1 D5 11 40 01 19 01 CD 80 06 06 03 :3F
C340: 3E FF D3 5D 77 23 10 FC D3 5F C9 06 0C 3E 18 D3 :49
C350: 5D 77 23 77 2B CD 80 06 10 F7 D3 5F C9 06 08 11 :0D
C360: 9A 04 D3 5C 1A 13 77 D3 5D 1A 13 77 CD 80 06 10 :A8
C370: F1 D3 5F C9 11 AA 04 18 03 11 BA 04 06 04 D3 5C :CE
C380: 1A 13 77 23 1A 13 77 2B D3 5D 1A 13 77 23 1A 13 :BA
C390: 77 2B CD 80 06 10 E7 D3 5F C9 00 18 08 3C 08 3C :87
C3A0: 00 3C 00 3C 42 7E C3 FF C3 C3 FC 00 FC 00 00 00 :78
C3B0: 3F FF 00 00 3F FF FC 00 FC 00 00 3F 00 3F 00 00 :F2
C3C0: FF FC 00 00 FF FC 00 3F 00 3F 11 10 A9 EB 06 0C :3B
C3D0: CD 6F 05 CD 78 06 08 10 F7 C9 06 04 3E 78 D3 5C :23
C3E0: 77 D3 5D 77 CD 80 06 10 F5 D3 5F C9 CD 80 06 3E :02
C3F0: FF D3 5C 77 D3 5D 77 D3 5F CD 80 06 3E FF D3 5C :3D
sum : 6A 8F 02 BE C3 D7 C6 86 D1 51 25 47 CA 80 8B 59 :5B

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: 77 D3 5D 77 D3 5F C9 11 1E 05 06 04 1A D3 5C 77 :17
C410: 13 1A D3 5D 77 CD 80 06 13 10 F1 D3 5F C9 3C 3C :AE
C420: FF E7 FF E7 3C 3C C5 ED 4B 8B 05 18 04 C5 01 03 :86
C430: 00 C5 D5 D3 5C ED 80 D1 C1 C5 D5 D3 5D ED 80 D1 :30
C440: C1 D5 D3 5E ED 80 D3 5F D1 C1 C9 C5 01 03 00 C5 :7F
C450: D5 D3 5C ED 80 D1 C1 D5 D3 5D ED 80 D3 5F D1 C1 :99
C460: C9 C5 D5 01 03 00 D3 5E ED 80 D3 5F D1 C1 C9 C5 :87
C470: D5 01 03 00 D3 5D ED 80 D3 5F D1 C1 C9 C5 D5 01 :CE
C480: 03 00 D3 5C ED 80 D3 5F D1 C1 C9 00 00 7E 32 8B :97
C490: 05 23 7E 23 F5 CD 26 05 CD 78 06 F1 30 20 F5 C9 :0D
C4A0: CD C0 05 D0 06 0C CD 4B 05 CD 78 06 D8 10 F7 C9 :84
C4B0: CD C0 05 D0 06 0C CD 7D 05 CD 78 06 D8 10 F7 C9 :B6
C4C0: E5 D5 21 00 C0 A7 ED 52 D1 E1 C9 3A 07 E0 A7 28 :EC
C4D0: 0F FE 10 30 28 06 0C CD 2D 05 CD 78 06 10 F8 C9 :A2
C4E0: 06 0C E5 21 00 C0 A7 ED 52 E1 30 05 CD 2D 05 18 :EB
C4F0: 06 C5 01 09 00 C9 C1 CD 78 06 10 E6 C9 CD C0 05 :3B
sum : 5F 4E 7D 53 2B 3E 06 1C 11 32 C0 F1 D8 DE 31 C7 :AA

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: D0 06 0C 18 14 06 0C E5 21 7E FE A7 ED 52 E1 D8 :41
C510: E5 21 00 C0 A7 ED 52 E1 D0 CD 2D 05 CD 78 06 D8 :7F
C520: 10 F7 C9 3A 07 E0 A7 28 0F FE 10 30 25 06 0C CD :11
C530: 61 05 CD 78 06 10 F8 C9 06 0C E5 21 00 C0 A7 ED :EE
C540: 52 E1 30 05 CD 61 05 18 03 23 23 23 CD 78 06 10 :7A
C550: E9 C9 CD C0 05 D0 06 0C 18 14 06 0C E5 21 7E FE :E6
C560: A7 ED 52 E1 D8 E5 21 00 C0 A7 ED 52 E1 D0 CD 61 :2A
C570: 05 CD 78 06 D8 10 F7 C9 E5 21 50 00 19 EB E1 C9 :FC
C580: D5 11 50 00 19 D1 C9 D5 11 A0 00 19 D1 C9 D5 50 :54
C590: 16 C0 2E 50 CD 12 09 19 D1 C9 C8 24 C8 24 2C 2C :25
C5A0: 2C 18 EB C5 D5 21 43 BC 11 50 00 3A 02 E0 A7 28 :35
C5B0: 04 47 19 10 FD 11 C0 03 3A 09 E0 A7 28 04 47 19 :9B
C5C0: 10 FD EB 3A 08 E0 6F 26 03 CD 12 09 19 D1 C1 C9 :0E
C5D0: D5 E5 2A FC 06 11 00 E0 7C BA 38 03 21 00 D8 23 :64
C5E0: 22 FC 06 ED 5F D5 5F 16 00 19 D1 7C BA 38 03 D6 :EB
C5F0: 08 67 7E 69 67 CD 12 09 4C E1 D1 C9 00 D8 21 00 :65
sum : 37 FC 84 E7 D6 B1 D5 76 BE 97 1D ED 45 96 78 2E :50

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: C0 06 03 0E C8 CD 5B 07 21 F0 C0 06 03 0E C2 CD :45
C610: 91 03 21 33 C0 06 1C 0E C8 CD 5B 07 21 23 C1 06 :DA
C620: 1C 0E C2 CD 91 03 21 B4 C2 06 1A 0E 32 C0 7F 03 :93
C630: 21 84 04 06 1A 0E 25 CD 7F 03 21 A4 E1 06 1A 0E :EF
C640: 1B CD 7F 03 21 A4 EB 06 1A 0E 11 CD 7F 03 21 24 :ED
C650: F3 06 1A 0E 1D CD 7F 03 C3 D3 07 E5 C5 06 01 16 :F1
C660: 88 1E 22 CD 29 03 C1 E1 E5 C5 05 23 10 FD 06 01 :49
C670: 16 EE 1E 8B CD 29 03 C1 E1 E5 C5 23 CD 87 06 CD :6C
C680: 87 06 05 05 79 06 08 4F 16 AA 1E 55 CD 29 03 C1 :2A
C690: E1 E5 C5 0E 04 16 EE 1E 8B CD 29 03 C1 E1 E5 CD :C7
C6A0: 4C 03 E1 E5 78 3D 85 6F CD 51 03 E1 E5 C5 79 06 :89
C6B0: 04 47 CD 80 06 10 FB C1 E5 C5 0E 04 16 88 1E 22 :04
C6C0: CD 29 03 C1 E1 E5 CD 56 03 E1 78 3D 85 6F CD 58 :58
C6D0: 03 E1 C9 21 A0 CE 11 60 08 0E 06 CD 08 10 21 16 :88
C6E0: D6 11 67 08 0E 07 CD 0B 10 21 36 D9 11 60 08 0E :E7
C6F0: 07 CD DB 10 21 56 DC 11 73 08 0E 07 CD DB 10 21 :8C
sum : 9F 97 19 1F 12 CA E8 80 DE F6 52 DE 1F AF CF 12 :65

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: 22 D6 11 79 08 0E 06 CD DB 10 21 42 D9 11 7F 08 :2A
C710: 0E 06 CD DB 10 CD 8B 08 21 56 E6 11 DC 08 0E 07 :93
C720: CD DB 10 21 36 ED 11 E8 08 0E 06 CD DB 10 21 B6 :A0
C730: F4 11 85 08 0E 07 CD DB 10 CD 13 10 21 8A F8 11 :03
C740: 5E 08 0E 07 CD DB 10 3A 47 E0 CD 43 10 21 40 E0 :F5
C750: 06 06 C5 E5 7E CD 4C 10 E1 23 C1 10 F5 C9 30 00 :20
C760: 4C 45 46 54 20 30 00 43 42 3A 4E 4F 00 54 42 3A :A7
C770: 4E 4F 00 53 42 3A 4E 4F 00 4D 49 3A 4E 4F 00 53 :C9
C780: 4C 3A 4E 4F 00 53 43 4F 52 45 00 21 36 E3 06 16 :F5
C790: 0E 07 E5 CD 5D 02 E1 3A 46 E0 A7 20 08 11 B8 08 :07
C7A0: 0E 07 C3 DB 10 3D 20 08 11 C4 08 0E 04 C3 DB 10 :C5
C7B0: 11 D0 08 0E 06 C3 DB 10 4E 4F 52 4D 4C 2E 41 52 :F4
C7C0: 4D 45 52 00 42 41 54 4C 45 2E 53 48 49 4C 44 00 :EE
C7D0: 53 55 50 45 52 2D 53 48 49 4C 44 00 42 41 52 52 :57
C7E0: 59 45 52 2D 3A 4E 4F 00 42 53 54 2E 46 55 45 4C :2A
C7F0: 3A 4E 4F 00 3A 1F E0 47 0E 06 21 AA CE C3 EC 10 :C3
sum : 9B AF CD 7A 84 11 0E F0 53 D6 52 C8 31 CA F9 71 :CC

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: D5 11 00 C0 E5 A7 ED 52 E1 30 05 CD 80 06 18 F4 :E6
C810: D1 C9 F5 C5 AF 06 08 CB 1D 30 01 84 1F 10 F8 CB :A0
C820: 1D 67 C1 F1 C9 C5 D5 06 07 11 00 40 0E 0C 7D 91 :D3
C830: 3B 04 6F 7A 83 5F CB 0A CB 09 10 F2 65 6B 01 C1 :14
C840: C9 3A 02 E0 CB 3F CB 3F 47 7C 90 30 01 AF 67 E5 :78
C850: CD 25 09 4D 7D 32 08 E0 7C 32 0A E0 06 00 2A 00 :A7
C860: E0 09 EB E1 6C CD 25 09 2C 7D 32 09 E0 07 07 07 :F5
C870: 17 30 02 06 01 4F EB 09 7E C9 F5 C5 E5 21 00 E1 :7B
C880: 06 0A 7E A7 28 0E 3C 28 0B E5 36 FF 23 23 23 23 :80
C890: 23 36 05 E1 C5 01 08 00 09 C1 10 E6 E1 C1 F1 C9 :29
C8A0: 21 00 E1 06 0A C5 E5 CD 8B 09 E1 C1 D0 23 23 23 :25
C8B0: 23 23 23 23 10 EE C9 7E A7 28 4D 3C 28 4A 23 E1 :E1
C8C0: 23 7E 93 28 08 3C 28 05 3C 28 02 18 3C 23 7E 92 :BA
C8D0: 28 08 3C 28 05 3C 28 02 18 2F 28 28 28 7E E6 10 :3B
C8E0: 28 06 3A 18 E0 3D 28 1F 7E 36 FF 23 23 23 23 :46
C8F0: 36 05 E6 0F CB 3F 3C 4F 06 00 CD EC 0F 3A 18 E0 :C5
sum : 9E D1 93 2C 67 36 43 91 5F 51 1F A6 87 45 16 B5 :AB

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: FE 06 28 05 3D 28 02 A7 C9 37 C9 D5 E5 CD A0 09 :38
C910: E1 D1 D0 D5 E5 CD 18 0A E1 D1 C9 EB CD 41 09 FE :A9
C920: 60 30 1A FE 0A 38 38 FE 0C 38 36 FE 10 38 30 FE :0E
C930: 26 38 2E FE 50 38 28 FE 58 38 34 18 24 FE 7E 38 :EC
C940: 1E FE 8F 38 1C 28 18 FE B3 28 14 FE CA 38 2D FE :57
C950: CC 28 0C FE CE 28 08 FE D0 38 12 FE E0 38 10 37 :7E
C960: C9 3A 18 E0 FE 04 28 F7 38 03 CD 4C 0E A7 C9 C6 :84
C970: 08 47 3A 18 E0 FE 05 20 E8 C3 BC 0C CD 2A 0D 30 :48
C980: EC CD CC 0D 18 E7 C5 D5 E5 EB CD 41 09 E1 D1 C1 :85
C990: FE 50 30 2A FE 0A 38 60 FE 0C 38 5E FE 10 38 58 :86
C9A0: FE 12 38 61 FE 16 38 6C FE 1A 38 63 FE 1C 38 64 :CA
C9B0: FE 1E 38 51 FE 24 38 5C FE 26 38 3E 18 3A FE 54 :99
C9C0: 38 43 FE 58 38 34 FE 5C 38 38 FE 60 38 2C FE 7E :48
C9D0: 38 26 FE 82 38 3E FE 84 38 28 FE 86 38 36 FE 88 :B1
C9E0: 38 18 FE 8C 38 2E FE 8E 28 18 38 0E FE 8F 28 08 :12
C9F0: FE 98 38 06 FE B2 28 0D 37 C9 79 FE 04 28 F9 FE :53
sum : AC 4C CB 59 FC 34 59 38 5F 1F CD 5C FA E5 D3 45 :7B

```



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CA00:	0C	28	F5	18	0F	79	A7	28	EF	FE	08	28	EB	18	05	79	:36
CA10:	E6	03	28	E4	AF	C9	E5	D5	C5	14	1C	EB	CD	41	09	C1	:DF
CA20:	D1	E1	FE	60	30	1D	A7	28	38	FE	03	28	34	FE	0A	38	:01
CA30:	2E	FE	0E	38	2C	1E	10	38	26	FE	30	38	24	FE	50	38	:1A
CA40:	1E	18	1E	FE	6E	38	18	FE	70	38	16	FE	7E	38	10	FE	:8E
CA50:	8F	28	0C	FE	B3	28	08	FE	E0	38	06	FE	FA	30	02	37	:21
CA60:	C9	AF	C9	3A	12	E0	6F	3A	13	E0	67	24	2C	CD	41	09	:07
CA70:	FE	60	30	3E	A7	CA	10	0C	FE	0A	DA	0A	0C	FE	0C	DA	:35
CA80:	F9	0B	FE	0E	DA	10	0C	FE	10	DA	FD	0B	FE	26	DA	F9	:ED
CA90:	0B	FE	30	DA	10	0C	FE	38	28	63	FE	30	28	5D	FE	3F	:F0
CAA0:	28	59	FE	4B	28	57	FE	50	38	60	FE	58	DA	C5	0C	C3	:F3
CAB0:	B9	0C	FE	68	38	54	FE	6E	28	56	FE	7A	38	49	FE	7E	:16
CAC0:	38	48	FE	80	38	35	FE	8F	38	2F	28	3E	FE	B3	28	3A	:D8
CAD0:	FE	C8	38	22	FE	CA	38	2A	FE	CC	28	2E	FE	CE	28	2A	:88
CAE0:	FE	E0	38	12	FE	F0	38	1A	FE	F3	38	0F	FE	FA	38	0D	:DD
CAF0:	FE	FD	38	07	18	07	CD	66	0C	AF	C9	18	0D	CD	91	0F	:A2
sum :	7C	B4	1C	5E	8A	24	23	CF	4B	F8	FC	4A	FF	61	C2	BB	:B0

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CB00:	18	08	CD	00	0E	18	03	CD	90	0C	AF	32	1A	E0	37	C9	:2A
CB10:	47	3A	10	E0	FE	02	28	25	3E	01	32	1A	E0	3A	48	E0	:8B
CB20:	D6	18	38	05	32	48	E0	18	E5	3A	47	E0	A7	28	E0	3D	:FD
CB30:	32	47	E0	3E	FF	32	48	E0	CD	43	10	18	01	C5	CD	43	:CE
CB40:	10	C1	3E	03	32	10	E0	3E	08	32	15	E0	78	FE	2C	28	:6B
CB50:	0B	FE	2A	38	07	3A	09	E0	3C	32	09	E0	CD	84	14	CD	:1E
CB60:	A3	06	22	16	E0	C9	47	FE	CD	CA	63	16	FE	CF	CA	63	:D9
CB70:	16	CD	0F	0D	7A	E6	08	C8	3A	46	E0	A7	CA	63	16	3D	:86
CB80:	20	0A	32	46	E0	C5	E5	CD	8B	08	E1	C1	78	C3	CC	0D	:42
CB90:	C9	11	10	00	A7	ED	52	28	7E	FE	50	38	08	FE	58	38	:95
CBA0:	13	FE	60	38	11	23	23	7E	FE	50	D8	FE	58	38	05	FE	:35
CBB0:	60	D0	18	02	C6	08	47	18	2A	C3	F9	0B	C5	E5	CD	4C	:2B
CBC0:	0E	E1	C1	18	1E	C6	08	47	3A	24	D0	3D	CA	F9	0B	3A	:6E
CBD0:	46	E0	A7	CA	F9	0B	3D	20	0A	32	46	E0	C5	E5	CD	8B	:5C
CBE0:	08	E1	C1	3A	24	D0	3D	28	24	78	FE	5C	30	0E	C6	08	:3F
CBF0:	CB	47	28	02	3D	2B	77	3C	23	77	18	11	11	10	00	C6	:01
sum :	BE	05	99	EF	A6	36	25	27	87	5C	C7	4D	EC	95	0E	E0	:D9

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CC00:	08	C8	47	28	04	A7	E0	52	3D	77	3C	19	77	AF	C9	F5	:19
CC10:	E5	21	50	D0	D6	90	38	0C	CB	27	5F	16	00	19	5E	23	:D1
CC20:	56	E1	F1	C9	11	00	00	18	F8	00	F5	FE	DC	D2	B4	0D	:74
CC30:	FE	D8	38	1A	3A	29	0D	FE	0A	30	07	3C	32	29	0D	C3	:3E
CC40:	B1	0D	AF	32	29	0D	3E	80	32	10	E0	C3	B4	0D	CD	0F	:22
CC50:	0D	7B	E6	80	28	1B	3A	08	E0	32	06	E0	3A	09	E0	32	:C0
CC60:	07	E0	D5	E5	7B	E6	10	F5	C4	F4	1F	F1	CC	4C	20	E1	:E8
CC70:	D1	3A	18	E0	5F	7A	E6	07	28	3A	3D	20	06	7B	3D	28	:6E
CC80:	30	18	31	3D	20	07	7B	FE	05	38	26	18	27	3D	28	21	:7E
CC90:	3D	20	07	7B	FE	02	28	1C	18	12	3D	20	07	7B	FE	03	:2D
CCA0:	28	12	18	08	3D	20	05	7B	FE	04	28	08	7B	FE	06	28	:10
CCB0:	03	F1	A7	C9	F1	37	C9	C5	0E	04	CD	D0	06	79	FE	03	:49
CCC0:	28	05	3E	E0	81	18	02	3E	E5	77	C1	C9	FE	90	D8	FE	:6E
CCD0:	94	30	07	36	8C	CD	AE	0F	18	72	FE	98	30	07	36	8D	:31
CCE0:	CD	AE	0F	18	67	CD	AE	0F	FE	B2	20	04	36	8E	18	5C	:9F
CCF0:	FE	B4	30	04	36	8F	18	54	FE	B8	38	74	FE	C0	DA	93	:A4
sum :	F6	19	BD	0D	46	89	87	02	2A	F0	48	06	56	B4	1C	FB	:BA

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CD00:	0E	FE	C4	20	05	CD	B7	0D	18	42	FE	C5	20	04	36	3D	:3A
CD10:	18	3A	FE	C6	20	05	CD	B7	0D	18	31	FE	C7	20	04	36	:34
CD20:	3F	18	29	FE	CA	30	04	36	8F	18	21	FE	00	D8	FE	04	:F2
CD30:	30	11	D6	10	77	CB	47	F5	CC	7A	09	F1	FE	C3	CC	91	:03
CD40:	0C	18	09	FE	D8	38	29	FE	E0	38	48	C9	CD	A3	06	CD	:CE
CD50:	00	09	EB	0E	07	21	50	E0	06	10	7E	A7	20	0B	71	23	:54
CD60:	73	23	72	CD	06	01	C3	42	1C	23	23	23	23	10	EB	C9	:4D
CD70:	E6	01	28	08	2B	3A	08	E0	3D	32	08	E0	3E	8F	77	23	:22
CD80:	77	CD	A3	06	11	E0	01	A7	ED	52	CD	00	09	EB	0E	25	:B9
CD90:	C3	55	0E	E6	03	28	28	3D	28	15	3D	20	0A	2B	3A	08	:AD
CDA0:	E0	3D	32	08	E0	18	18	28	3A	08	E0	3D	32	08	E0	3A	:45
CDB0:	09	E0	A7	28	04	3D	32	09	E0	11	10	00	A7	ED	52	3E	:59
CDC0:	8F	77	23	77	11	0F	00	19	77	23	77	CD	A3	06	18	BA	:32
CDD0:	F5	E5	CD	A3	06	CD	8B	02	CD	A1	01	E1	F1	FE	C8	28	:09
CDE0:	4C	FE	C2	28	48	FE	E0	D8	FE	E6	38	46	20	0E	3A	46	:49
CDFO:	0C	FE	02	28	32	3E	01	32	46	E0	18	38	FE	E7	20	07	:30
sum :	CD	3D	94	5B	FF	D6	F2	2C	76	93	0C	B1	A1	10	91	88	:7C

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CE00:	3E	02	32	46	E0	18	30	FE	E8	20	15	3A	47	E0	C6	0A	:2C
CE10:	FE	64	38	02	3E	63	32	47	E0	E5	CD	43	10	E1	18	07	:9B
CE20:	CD	AE	0F	FE	ED	30	03	36	8F	C9	36	6F	C9	CD	3E	0F	:8E
CE30:	18	F5	CD	49	0F	18	F0	E5	CD	8B	08	E1	18	E9	3A	1F	:BA
CE40:	E0	3C	32	1F	E0	CD	F4	08	C9	E5	D6	E0	4F	06	00	21	:F0
CE50:	40	E0	09	FE	03	30	2A	03	0C	E0	CB	3F	CB	3F	CB	3F	:BE
CE60:	C6	05	4F	47	CD	00	06	78	81	3C	3C	FE	1E	38	02	3E	:09
CE70:	1E	47	CD	84	0F	E1	C9	20	04	06	02	18	02	06	01	CD	:89
CE80:	84	0F	E1	C9	7E	80	FE	64	38	02	3E	63	77	CD	4C	10	:18
CE90:	C9	FE	38	20	07	0E	C4	06	8F	C3	A3	0F	FE	4B	C0	0E	:1C
CEA0:	C6	06	8F	E5	11	50	00	A7	E0	52	71	E1	70	C9	F5	C5	:CC
CEB0:	D5	E5	D6	90	30	01	AF	CB	3F	CB	3F	CB	3F	07	4F	06	:7A
CEC0:	00	21	D0	0F	09	4E	23	46	CD	EC	0F	E1	D1	C1	F1	C9	:B5
CED0:	08	00	10	00	15	00	20	00	30	00	50	00	80	00	00	01	:4E
CEE0:	50	01	00	02	00	03	50	03	00	04	00	05	3A	0D	E0	81	:5A
CEF0:	27	32	0D	E0	3A	0E	E0	88	27	32	0E	E0	3A	0F	E0	0E	:74
sum :	8C	BD	0B	C6	F7	AF	1C	E7	95	64	FD	E6	5B	BF	25	EC	:CA

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CF00:	00	89	27	32	0F	E0	30	0B	3E	99	32	0D	E0	32	0E	E0	:22
CF10:	32	0F	E0	21	7E	F8	DD	21	0F	E0	06	03	0E	07	C5	DD	:65
CF20:	7E	00	CB	3F	CB	3F	CB	3F	CB	3F	47	CD	EC	10	23	23	:FC
CF30:	DD	7E	00	E6	0F	47	0E	07									



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D400: 0E 15 18 71 CD 28 15 18 55 CD 28 15 18 67 CD 00 :49
D410: 06 3A 13 E0 FE 2D 00 3C 32 13 E0 CD 63 08 08 3A :DC
D420: 13 E0 3D 32 13 E0 18 18 CD 00 06 3A 13 E0 FE 09 :5C
D430: 08 3D 32 13 E0 CD 63 08 08 3A 13 E0 3C 32 13 E0 :DB
D440: 08 08 CB 6F 28 03 CB 77 C0 3A 47 E0 A7 C8 3A 10 :64
D450: E0 3D C0 3E 02 32 10 E0 3E 14 32 15 E0 C9 3A 12 :CD
D460: E0 FE 01 D8 3D 32 12 E0 CD 63 08 08 3A 12 E0 3C :93
D470: 32 12 E0 18 CB 3A 12 E0 FE 2D 00 3C 32 12 E0 CD :58
D480: 63 08 08 3A 12 E0 3D 32 12 E0 18 B4 3A 10 E0 A7 :70
D490: C8 FE 01 28 18 FE 02 20 1D 3A 15 E0 A7 28 09 3D :88
D4A0: 32 15 E0 CD 84 14 18 3D 3E 01 32 10 E0 CD 84 14 :A7
D4B0: CD 8A 14 C3 27 14 3A 15 E0 A7 CA 63 16 3D 32 15 :06
D4C0: E0 CB 3F A7 20 05 21 30 A6 18 13 3D 20 05 21 E0 :38
D4D0: A5 18 08 3D 20 05 21 90 A5 18 03 21 40 A5 ED 58 :E9
D4E0: 16 E0 C3 A0 05 3A 11 E0 A7 20 03 3C 18 0F 3D 20 :13
D4F0: 04 3E 03 18 08 3D 20 03 AF 18 02 3E 02 32 11 E0 :F1
sum : 95 6A E3 C1 12 2A 63 D5 E3 F2 B9 E4 0E 66 E5 66 :48

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D500: 3A 12 E0 6F 3A 13 E0 67 CD 9A 06 22 16 E0 E5 21 :8A
D510: 00 90 3A 46 E0 A7 28 04 11 E0 01 19 3A 11 E0 A7 :A0
D520: 28 0F 3D 20 05 11 40 01 18 06 3D 20 08 11 A0 00 :1F
D530: 19 01 C3 A0 05 3A 46 E0 A7 20 05 21 A0 A4 18 F1 :EC
D540: 21 F0 A4 18 EC 3A 10 E0 A7 C8 FE 03 C8 CD 63 08 :56
D550: D8 CD 84 14 3A 13 E0 3C FE 2E 30 07 D0 32 13 E0 :FE
D560: C3 27 14 CD 84 14 AF 32 10 E0 3E 0A 32 15 E0 ED :90
D570: 58 16 E0 0E 07 C3 55 0E 3A 18 E0 A7 C0 3A 10 E0 :4F
D580: 3D C0 DB 09 CB 77 28 41 CB 4F CA E7 17 CB 57 CA :5A
D590: B8 17 CB 5F CA 89 17 CB 67 CA FF 16 CB 6F CA 45 :80
D5A0: 17 DB 02 CB 5F 20 0E DB 05 CB 47 20 1C CB 57 CA :66
D5B0: E7 17 C3 89 17 DB 05 CB 47 20 08 CB 57 C2 45 17 :8B
D5C0: C3 B8 17 CB 57 CA FF 16 C9 3E 01 CD ED 16 A7 20 :32
D5D0: 07 1C 1C 14 3E 02 18 0C 3D 20 05 1C 3E 04 18 E0 :93
D5E0: 1C 14 14 AF 21 20 E0 CD 39 17 C3 25 18 32 18 E0 :58
D5F0: CD 37 01 3A 12 E0 5F 3A 13 E0 57 3A 11 E0 C9 3A :42
sum : 38 64 E9 00 A8 F0 2A 83 5C E7 CD 67 2B E7 40 9F :32

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D600: 40 E0 A7 C8 3D 32 40 E0 CD 61 10 3E 02 CD ED 16 :6C
D610: 1C 14 A7 20 06 06 01 0E 03 18 0D 3D 20 06 06 03 :A6
D620: 0E 05 18 04 06 01 0E 07 78 21 20 E0 CD 39 17 79 :7A
D630: 21 28 E0 CD 39 17 C3 BD 19 36 01 23 77 23 73 23 :69
D640: 72 23 36 00 C9 3A 41 E0 A7 C8 3D 32 41 E0 CD 66 :21
D650: 10 3E 03 CD ED 16 A7 20 05 1C 1C 14 18 09 3D 20 :87
D660: 03 1C 18 03 1C 14 14 21 20 E0 3E 02 CD 39 17 21 :10
D670: 28 E0 3E 04 CD 39 17 21 30 E0 AF CD 39 17 21 38 :8D
D680: E0 3E 06 CD 39 17 C3 BD 19 3A 42 E0 A7 C8 3D 32 :14
D690: 42 E0 CD 68 10 3E 04 CD 9D 17 C3 25 18 CD ED 16 :FD
D6A0: A7 20 05 14 3E 02 18 0A 3D 20 04 3E 04 18 03 1C :1C
D6B0: 1C AF 21 20 E0 C3 39 17 3A 43 E0 A7 C8 3D 32 43 :7D
D6C0: E0 CD 70 10 3E 05 CD ED 16 A7 20 06 1C 14 3E 02 :7D
D6D0: 18 0C 3D 20 05 14 3E 04 18 04 1C 1C 14 AF 21 20 :34
D6E0: E0 CD 39 17 C3 24 18 3A 44 E0 A7 C8 3D 32 44 E0 :5F
D6F0: CD 75 10 3E 06 CD 9D 17 C3 24 18 3A 18 E0 A7 C8 :8A
sum : C2 86 C4 7E 94 11 00 E1 BF D7 6B A1 D5 27 68 05 :18

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D700: 3D CA 25 18 3D CA BD 19 3D CA BD 19 3D CA 25 18 :42
D710: 3D CA 24 18 3D CA 24 18 C9 7E 47 23 5E 23 56 23 :37
D720: 4E 0C 71 28 C9 21 21 E0 CD 19 18 FE 02 20 51 C5 :15
D730: D5 E5 CD A0 09 E1 D1 C1 D2 16 19 15 15 3E 02 BA :C8
D740: D2 14 19 CD 0B 0A D2 14 19 3A 18 E0 3D 28 13 D5 :5F
D750: 1C CD 0B 0A D1 D2 14 19 D5 1C 1C CD 0B 0A D1 D2 :60
D760: 14 19 CD A5 19 3A 24 E0 FE 04 D8 3E 06 84 67 CD :CC
D770: 9A 06 3A 18 E0 3D CA CD 03 06 03 0E 07 C3 F8 02 :C4
D780: FE 04 20 4E C5 D5 E5 CD A0 09 E1 D1 C1 D2 41 19 :04
D790: 10 10 3E 2F BB DA 3F 19 CD 0B 0A D2 3F 19 3A 18 :F2
D7A0: E0 3D 28 13 D5 14 CD 0B 0A D1 D2 3F 19 D5 14 14 :1B
D7B0: CD 0B 0A D1 D2 3F 19 CD 8E 19 3A 24 E0 FE 04 D8 :69
D7C0: 3E 05 85 6F 3A 18 E0 3D CA 1D 03 06 03 0E 0C C3 :76
D7D0: F8 02 C5 D5 E5 CD A0 09 E1 D1 C1 D2 72 19 1C 1C :F7
D7E0: 3E 2E BB DA 74 19 CD 0B 0A D2 74 19 D5 14 14 CD :0B
D7F0: 28 13 D5 14 CD 0B 0A D1 D2 74 19 D5 14 14 CD 0B :0B
sum : 9D 36 1C 25 A8 F4 08 CF 20 09 8C 14 C3 D5 79 74 :D5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D800: 0A D1 D2 74 19 CD 8E 19 3A 24 E0 FE 04 D8 7D 06 :19
D810: 06 6F 18 B0 14 14 EB CD 9A 06 3A 24 E0 3D 28 4A :AA
D820: FE 03 38 02 3E 03 CB 27 CB 27 CB 27 4F 3A 18 E0 :03
D830: 3D 20 05 CD 0F 03 18 32 06 03 CD F8 02 18 2B 1C :BA
D840: 1C EB CD 9A 06 3A 24 E0 3D 28 1F FE 03 38 02 3E :AF
D850: 03 47 3A 18 E0 3D 28 09 CB 20 0E 0C CD F8 02 18 :CE
D860: 09 E5 CD 1D 03 E1 23 23 10 F7 AF 32 18 E0 32 20 :34
D870: E0 C9 1C 1C 3A 24 E0 3D 28 F0 FE 03 38 02 3E 03 :F0
D880: F5 CB 27 47 78 90 5F EB CD 9A 06 F1 18 C3 2B 73 :5A
D890: EB CD 9A 06 E5 3A 18 E0 3D 20 05 CD 01 04 18 03 :8E
D8A0: CD 4B 04 E1 C9 72 EB E5 D5 CD 9A 06 3A 18 E0 3D :B9
D8B0: 20 05 CD F2 03 18 03 CD 30 04 D1 E1 C9 21 20 E0 :9F
D8C0: E5 06 04 C5 E5 CD E1 19 E1 01 08 00 09 C1 10 F3 :17
D8D0: E1 11 08 00 06 04 7E A7 C0 19 10 FA AF 32 18 E0 :E5
D8E0: C9 7E A7 C8 23 CD 19 18 CB 47 CA 83 1A F5 CD 68 :7D
D8F0: 1A F1 FE 01 20 1C 1C 1C 15 15 3E 02 BA D2 65 1A :F3
sum : C9 B1 5A 8C F7 71 A4 F9 75 84 22 A4 FD 33 F9 80 :CD

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D900: 3E 2E BB DA 65 1A CD 0B 0A D2 65 1A CD 7C 1A C3 :D9
D910: 13 04 FE 03 20 19 1D 1D 15 15 3E 02 BA 30 46 3E :63
D920: 2F BB 38 41 CD 0B 0A 30 3C CD 7C 1A C3 0E 04 FE :E7
D930: 05 20 19 1D 1D 14 14 3E 2E BA 38 29 3E 2F BB 38 :87
D940: 24 CD 0B 0A 30 1F CD 7C 1A C3 13 04 1C 1C 14 14 :F2
D950: 3E 2E BA 38 10 3E 2E BB 38 0B CD 0B 0A 30 06 CD :8D
D960: 7C 1A C3 0E 04 2B 2B 2B 36 00 C9 79 3D C8 E5 D5 :23
D970: C5 EB CD 9A 06 CD 0D 02 C1 D1 E1 C9 72 2B 73 EB :00
D980: C3 9A 06 A7 20 1A 79 3D C4 01 1B 1C 1C 3E 2E BB :39
D990: DA F2 1A CD 0B 0A D2 F2 1A CD 7C 1A 2B C3 01 04 :FC
D9A0: FE 02 20 17 79 3D C4 16 1B 15 15 3E 02 BA 30 42 :78
D9B0: CD 0B 0A 30 3D CD 7C 1A C3 F2 03 FE 04 20 17 79 :1C
D9C0: 3D C4 F8 1A 1D 1D 3E 2F BB 38 27 CD 0B 0A 30 22 :08
D9D0: CD 7C 1A C3 01 04 79 3D C4 10 1B 14 14 3E 2D BA :1D
D9E0: 38 10 CD 0B 0A 30 0B 72 2B 73 15 EB CD 9A 06 C3 :A5
D9F0: F2 03 2B 2B 2B 36 00 C9 C5 D5 E5 EB CD 9A 06 18 :64
sum : C4 F9 B3 F3 ED 5C 58 00 FD 72 CC D9 63 7F 70 09 :73

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DA00: 08 C5 D5 E5 EB CD 9A 06 2B CD 1D 03 E1 D1 C1 C9 :33
DA10: C5 D5 E5 15 18 03 C5 D5 E5 EB CD 9A 06 CD 0D 03 :63
DA20: E1 D1 C1 C9 21 20 E0 7E A7 C8 23 CD 19 18 F5 79 :D9
DA30: FE 04 20 10 3A 18 E0 FE 06 20 09 C5 D5 E5 CD 6E :48
DA40: 27 E1 D1 C1 F1 A7 20 3D 79 3D C4 26 1C 1C 1C 3E :C1
DA50: 2E BB DA FB 18 CD 0B 0A D2 FB 18 3A 18 E0 FE 05 :D8
DA60: 28 13 D5 14 CD 0B 0A D1 D2 FB 18 D5 14 14 CD 0B :94
DA70: 0A D1 D2 FB 18 CD 7C 1A 2B 3A 18 E0 FE 05 CA 74 :C4
DA80: 04 2B C3 CA 04 FE 02 20 39 79 3D C4 03 1C 15 15 :0C
DA90: 3E 02 BA 30 66 CD 0B 0A 30 61 3A 18 E0 FE 05 28 :60
DAA0: 13 D5 1C CD 0B 0A D1 D2 FB 18 D5 1C 1C CD 0B 0A :8E
DAB0: D1 D2 FB 18 CD 7C 1A 3A 18 E0 FE 05 CA 5D 04 C3 :3F
DAC0: CA 04 79 3D C4 16 1C 1D 1D 3E 2F BB 38 2D CD 0B :19
DAE0: 0A 30 28 3A 18 E0 FE 05 28 13 D5 14 CD 0B 0A D1 :6E
DAF0: D2 FB 18 D5 14 14 CD 0B 0A D1 D2 FB 18 CD 7C 1A :E3
sum : 39 0A 1D CA 89 79 28 F0 93 CE 4C BA 36 19 9D A7 :3E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DB00: 18 E0 C9 C5 D5 E5 EB CD 9A 06 3A 18 E0 FE 05 20 :ED
DB10: 2A CD 91 02 18 28 C5 D5 E5 EB CD 9A 06 3A 18 E0 :03
DB20: FE 05 20 17 18 0F C5 D5 E5 EB CD 9A 06 2B 3A 18 :B5
DB30: E0 FE 05 20 05 CD 97 02 18 04 2B CD F4 02 E1 D1 :2A
DB40: C1 C9 21 50 E0 06 10 C5 E5 CD 55 1C E1 01 04 00 :8F
DB50: 09 C1 10 F3 C9 7E A7 C8 F5 E6 0F 47 F1 E6 F0 4F :CA
DB60: 20 35 78 3D 77 CD E5 1C C8 3D 20 05 21 90 A0 18 :E2
DB70: 23 3D 20 05 21 40 A0 18 18 3D 20 05 21 F0 9F 18 :E3
DB80: 13 3D 20 05 21 A0 9F 18 0B 3D 20 05 21 50 9F 18 :82
DB90: 03 21 90 A0 C3 A0 05 78 3D 77 CD E5 1C C8 F5 7E :F1
DBA0: C6 20 77 F1 E6 01 28 19 21 20 A2 01 C0 A2 D5 CD :5E
DBB0: D8 1C D1 21 C0 03 19 D8 EB 21 70 A2 01 10 A3 18 :84
DBC0: 17 21 60 A3 01 00 A4 D5 CD 0B 1C D1 21 C0 03 19 :44
DBD0: D8 EB 21 B0 A3 01 50 A4 D5 C5 CD A0 05 E1 D1 13 :FD
DBE0: 13 13 C3 A0 05 F5 23 5E 23 56 E5 05 EB 79 FE 20 :B9
DBF0: 30 05 CD 8B 02 18 03 CD A3 02 D1 21 A0 00 19 EB :B2
sum : 13 6A 51 B8 80 CC 47 5F F5 F7 41 7A A3 B0 62 1A :EE

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DC00: E1 72 28 73 28 38 02 F1 C9 28 36 00 23 F1 AF C9 :FD
DC10: 4D 21 90 E0 06 04 7E A7 28 07 23 23 23 23 10 F6 :FE
DC20: C9 79 E6 0F E5 6F 26 03 CD 12 09 4D 06 01 E1 3A :08
DC30: 0B E0 FE 08 38 01 04 70 23 23 71 23 36 00 C9 00 :77
DC40: 21 90 E0 06 04 C5 E5 CD 53 1D E1 01 04 00 09 C1 :32
DC50: 10 F3 C9 7E A7 C8 E5 AF 32 3F 1D CD 81 10 E1 7E :A5
DC60: A7 C8 E5 23 23 23 7E E6 7F 47 E1 3A 13 E0 4F 90 :D4
DC70: 38 05 FE 0F D0 18 05 78 91 FE 06 D0 3E 01 32 3F :C4
DC80: 10 E5 4E 23 46 23 5E 23 56 E1 CD D9 1E 3A 3F 10 :EE
DC90: A7 20 10 7A E6 80 28 07 7A E6 7F C9 57 18 04 7A :EE
DCA0: C6 80 57 78 A7 28 03 1D 18 01 1C 3E 30 BA 20 06 :87
DCB0: 36 00 23 36 00 C9 C5 D5 E5 EB 7C E6 7F 67 CD 41 :18
DCC0: 09 E1 D1 C1 23 FE 09 28 0C 78 A7 28 05 36 00 1C :78
DCD0: 18 03 36 01 1D 23 73 23 72 05 C5 21 00 00 7A E6 :85
DCE0: 80 28 07 21 A0 00 7A E6 7F 57 EB CD 9A 06 19 EB :02
DCF0: 79 3D 20 10 21 80 A6 D5 CD A0 05 D1 13 13 13 21 :9F
sum : EC 0A 31 5E C0 A9 E1 07 0D FF F8 8B 2E D5 AA F3 :05

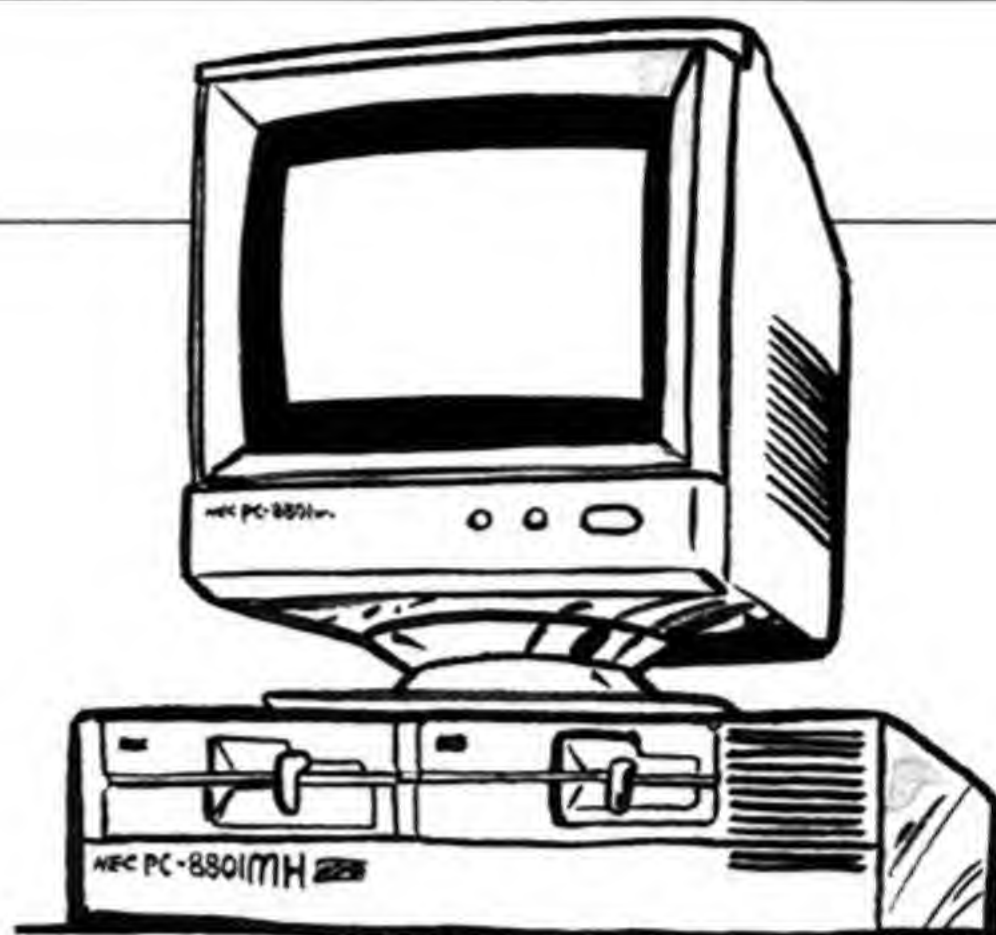
```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DD00: 0A A6 18 0E 21 20 A7 D5 CD A0 05 D1 13 13 13 21 :F6
DD10: 70 A7 CD A0 05 C1 D1 7A E6 7F 57 3A 13 E0 47 92 :57
DD20: 38 05 FE 03 D0 18 05 7A 90 FE 03 D0 3A 12 E0 47 :79
DD30: 93 38 05 FE 06 D0 18 05 7B 90 FE 02 D0 3A 10 E0 :C6
DD40: A7 C8 3D C2 63 16 79 3D C2 63 16 D5 CD 84 14 CD :DF
DD50: 68 01 D1 3A 13 E0 BA 38 1D 28 30 3D 28 2D 3D 28 :C5
DD60: 2A 3A 13 E0 C6 06 FE 20 30 05 32 13 E0 18 49 3E :47
DD70: 2D 32 13 E0 18 42 3A 13 E0 D6 06 FE 09 38 05 32 :2B
DD80: 13 E0 18 3A 3E 09 32 13 E0 18 2D 3A 12 E0 BB 38 :0F
DD90: 15 3A 12 E0 C6 06 FE 20 30 05 32 12 E0 18 19 3E :00
DDA0: 2D 32 12 E0 18 12 3A 12 E0 D6 06 38 05 32 12 E0 :E4
DDB0: 18 06 AF 32 12 E0 18 00 3A 12 E0 5F 3A 13 E0 57 :18
DDC0: CD 16 08 3D 0E CD 27 14 3E 02 32 10 E0 3E 0C 32 :12
DDD0: 15 E0 C9 CD 27 14 C3 63 16 C5 D5 E5 21 00 00 7A :1C
DDE0: E6 80 28 07 21 A0 00 7A E6 7F 57 EB CD 9A 06 19 :FD
DDF0: E5 CD 8B 02 E1 23 23 23 CD 8B 02 E1 D1 C1 C9 3A :59
sum : 8B 54 8E 97 B5 AC 8F E9 DE E9 80 A4 DE 16 8A EB :31

```





```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DE00: 12 E0 3C 6F 3A 13 E0 3C 67 7D B8 DA 54 1F 7C B9 :24
DE10: 38 21 7D 90 6F 7C 91 67 07 5F 7D B8 38 03 16 00 :38
DE20: C9 7D BC 38 03 16 0F C9 7D 07 BC 38 03 16 0E C9 :93
DE30: 16 0D C9 7D 90 6F 79 94 67 07 5F 7D B8 38 03 16 :CB
DE40: 01 C9 7D BC 38 03 16 02 C9 7D 07 BC 38 03 16 03 :83
DE50: C9 16 04 C9 7C B9 38 21 78 95 6F 7C 91 67 07 5F :90
DE60: 7D B8 38 03 16 09 C9 7D BC 38 03 16 0A C9 7D 07 :3C
DE70: BC 38 03 16 08 C9 16 0C C9 78 95 6F 79 94 67 07 :C3
DE80: 5F 7D B8 38 03 16 08 C9 7D BC 38 03 16 07 C9 7D :90
DE90: 07 BC 38 03 16 06 C9 16 05 C9 1E 01 18 02 1E 00 :1E
DEA0: 3A 13 E0 91 38 06 16 0C CD D2 1F D0 7B A7 20 06 :F4
DEB0: 16 04 CD D2 1F D0 3A 12 E0 90 38 04 16 00 18 02 :D0
DEC0: 16 08 3A 13 E0 91 28 08 3D 28 05 3C 28 02 37 C9 :DD
DED0: A7 C9 3A 12 E0 90 28 08 3C 28 05 3C 28 02 37 C9 :2B
DEE0: A7 C9 C5 CD FF 1E 7A A7 28 07 ED 5F E6 01 28 01 :CB
DEF0: 15 C1 18 20 3A 07 E0 FE 03 D8 FE 0D D0 CD 0D 22 :DF
sum : 5B 08 EB 02 7A DA F1 5E EB C2 00 C4 5B B9 66 42 :20

```

## リスト7-3 GAMMA 5 メインプログラム2 “LIST 2”

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: C5 CD FF 1E 7A A7 28 07 ED 5F E6 01 28 01 15 C1 :31
C010: CD 0C 21 D8 21 A0 E0 1E 06 7E A7 20 17 36 01 23 :01
C020: 72 23 70 23 0C 71 E5 68 61 CD 9A 06 E5 C1 E1 23 :6A
C030: 71 23 70 C9 D5 11 08 00 19 D1 1D 20 DC C9 6F E5 :DB
C040: C5 CD 9A 1F C1 E1 D8 7D BA C0 18 17 3A 07 E0 FE :0A
C050: 03 D8 FE 0D D0 CD 0D 22 0C C5 CD 9E 1F C1 D8 CD :73
C060: C0 21 D8 21 D0 E0 1E 06 C3 19 20 3A 10 E0 A7 C8 :43
C070: CD 0D 22 2A 04 E0 3A 13 E0 3C 91 38 14 D6 03 30 :59
C080: 10 3A 12 E0 C6 06 90 D8 D6 06 38 37 D6 05 D0 18 :7E
C090: 38 3A 12 E0 C6 02 90 D8 D6 03 D0 3A 13 E0 C6 04 :37
COA0: 91 D8 D6 05 30 02 18 05 D6 05 D0 18 0C 11 C0 03 :36
COB0: A7 ED 52 EB 21 E0 A8 18 1A 11 C0 03 19 EB 21 B0 :55
COC0: A8 18 10 2B 2B 2B EB 21 80 A8 18 07 23 23 23 EB :F8
C0D0: 21 50 A8 D5 CD 80 05 3E 05 D3 55 D3 59 06 03 C5 :05
COE0: ED 58 16 E0 21 C0 A7 D5 CD B0 05 CD 46 02 D1 21 :24
COF0: F0 A7 D5 CD 80 05 CD 46 02 D1 21 20 A8 CD B0 05 :3F
sum : F3 49 81 B6 87 C1 76 8C C6 70 05 C1 F5 18 E6 54 :00

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C100: CD 46 02 C1 10 D9 3E 01 D3 55 D3 59 D1 EB CD 8B :66
C110: 02 C3 63 16 3A 07 E0 FE 03 D8 FE 0D D0 CD 0D 22 :0F
C120: C5 CD FF 1E 06 02 7A D6 0F 20 04 16 00 18 01 14 :7D
C130: 10 F4 C1 3E 06 1E 06 21 A0 E0 F5 C5 CD 19 20 C1 :4F
C140: 7A A7 20 04 16 0F 18 01 15 F1 3D 20 E8 C9 3A 07 :D8
C150: E0 FE 03 D8 FE 0D D0 3A 10 E0 3D C0 CD 0D 22 0C :C3
C160: C5 0E 02 CD D0 06 0D C1 28 0E 3A 12 E0 B8 38 04 :9C
C170: 16 00 18 10 16 08 18 0C 3A 13 E0 B9 38 04 16 0C :C4
C180: 18 02 16 04 CD C0 21 D8 0D 05 7A A7 20 05 04 04 :1A
C190: 04 18 15 FE 04 20 05 0D 0D 0D 18 0C FE 08 20 05 :CE
C1A0: 05 05 05 18 03 0C 0C 0C 3E 2E B8 D8 B9 D8 7A FE :53
C1B0: 08 30 05 21 80 E1 18 03 21 88 E1 1E 01 C3 19 20 :7F
C1C0: 7A FE 02 20 04 1E 03 18 3A FE 06 20 04 1E 06 18 :75
C1D0: 32 FE 0A 20 04 1E 0C 18 2A FE 0E 20 04 1E 09 18 :39
C1E0: 22 FE 02 30 04 1E 01 18 1A FE 06 30 04 1E 02 18 :17
C1F0: 12 FE 0A 30 04 1E 04 18 0A FE 0E 30 04 1E 08 18 :10
sum : E2 C4 AF C7 B4 6F 09 52 0D DF B1 35 23 9B 75 2C :CB

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C200: 02 1E 01 3A 82 22 A3 20 02 37 C9 A7 C9 3A 02 E0 :50
C210: 6F CD 25 09 3A 07 E0 3D 4F 87 81 85 4F 3A 06 E0 :13
C220: 47 87 80 3C 47 C9 47 FE E0 D0 3A 1D E0 A7 C0 78 :A5
C230: FE 80 CA 68 20 ED 5F FE 40 D8 78 CD 0F D0 78 E6 :27
C240: 40 C8 CB 3A CB 3A CB 3A CB 3A 3A 0B E0 CB 3F FE :49
C250: 0F 38 02 3E 0F 82 3C 57 0E 32 CD D0 06 7A 91 D8 :71
C260: 7B E6 0F 32 82 22 78 E6 30 C8 FE 10 CA F4 1F FE :88
C270: 20 CA 4C 20 78 FE 9C DA 14 21 FE A0 D2 14 21 C3 :0F
C280: 4E 21 00 21 A0 E0 06 06 C5 E5 CD 96 22 E1 01 08 :35
C290: 00 09 C1 10 F3 C9 7E A7 C8 23 CD 83 23 4E 23 5E :E8
C2A0: 23 56 79 FE 04 30 21 A7 20 05 1C 1C C3 2C 23 30 :98
C2B0: 20 06 1C 1C 15 C3 2C 23 3D 20 07 1C 1C 15 15 C3 :0E
C2C0: 2C 23 1C 15 15 C3 2C 23 FE 08 30 22 D6 04 20 05 :FE
C2D0: 15 15 C3 2C 23 3D 20 06 1D 15 15 C3 2C 23 3D 20 :55
C2E0: 07 1D 15 15 C3 2C 23 1D 10 15 C3 2C 23 FE 0C :E8
C2F0: 30 1E D6 08 20 04 1D 1D 18 32 30 20 05 1D 1D 18 :84
sum : A9 CB C0 5D 10 1E AD 8A C8 54 53 BA E0 4C 27 60 :D2

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C300: 18 2A 3D 20 06 1D 1D 14 14 18 21 1D 14 14 18 1C :89
C310: D6 0C 20 04 14 14 18 14 3D 20 05 1C 14 14 18 0C :24
C320: 3D 20 06 1C 1C 14 14 18 03 1C 14 14 CD 90 23 D0 :72
C330: C3 07 05 21 D0 E0 06 06 C5 E5 CD 46 23 E1 01 08 :76
C340: 00 09 C1 10 F3 C9 7E A7 C8 23 CD 83 23 4E 23 5E :E8
C350: 23 56 79 A7 20 05 1C 1C 1C 18 15 FE 04 20 05 15 :7B
C360: 15 15 18 0C FE 08 20 05 1D 1D 10 18 03 14 14 14 :27
C370: C5 CD 90 23 C1 D0 79 A7 CA EC 04 FE 08 CA EC 04 :70
C380: C3 DA 04 E5 23 23 23 5E 23 56 E8 CD D0 02 E1 C9 :07
C390: 05 E5 78 FE 30 30 41 7A FE 30 30 3C 3A 10 E0 A7 :89
C3A0: 28 1D 3A 12 E0 93 28 06 3C 28 03 3C 20 11 3A 13 :53
C3B0: E0 92 28 06 3C 28 03 3C 20 05 CD E1 23 18 19 CD :37
C3C0: 86 0A 30 14 E1 D1 72 2B 73 23 23 E5 E8 CD 9A 06 :19
C3D0: EB E1 73 23 72 EB 37 C9 E1 D1 2B 28 2B 36 00 A7 :CF
C3E0: C9 CD 2D 02 3A 45 E0 A7 28 07 3D 32 45 E0 C3 7A :CB
C3F0: 10 11 D0 E0 A7 ED 52 38 20 3A 46 E0 A7 28 1A 3A :92
sum : D5 D5 CB 5B 7B C7 EC A2 FD 65 C6 72 A6 2B 07 3C :4E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: 11 E0 A7 20 06 79 FE 0C 20 0F C9 3D 20 05 79 A7 :BB
C410: 20 07 C9 79 FE 08 20 01 C9 CD 63 16 C9 3E 03 32 :0B
C420: 1F E0 21 40 E0 3E 14 77 23 77 3E 0A 23 77 3E 05 :C8
C430: 23 77 3E 03 23 77 3E 1E 32 47 E0 C9 3A 24 D0 A7 :C8
C440: 20 03 C3 FD 01 3D 20 0B 3A 40 D0 FE 12 CA FD 01 :6E
C450: C3 05 02 3D C2 21 02 C3 15 02 3A 0B E0 CB 3F 3C :31
C460: FE 06 38 02 3E 06 32 18 20 32 67 20 C9 DB 01 3C :86
C470: 28 FB C9 00 00 11 00 80 21 00 C0 01 00 10 ED B0 :0C
C480: 18 03 CD 0A 26 3A 25 D0 32 0B E0 21 00 C0 06 50 :9B
C490: 0E C8 CD 5D 02 21 00 E0 36 00 11 01 E0 01 00 02 :2E
C4A0: ED B0 CD 1D 24 CD FE 06 3A 1F E0 A7 CA 82 24 3D :09
C4B0: 32 1F E0 CD F4 08 3A 47 E0 FE 1E 30 08 3E 1E 32 :3D
C4C0: 47 E0 CD 43 10 AF 32 46 E0 CD 8B 08 21 00 80 11 :60
C4D0: 00 C0 01 00 10 ED B0 3A 0B E0 3C 32 0B E0 CD 5A :13
C4E0: 24 AF 32 18 E0 21 20 E0 36 00 11 21 E0 01 1F 00 :86
C4F0: ED B0 21 50 E0 36 00 11 51 E0 01 40 01 ED B0 21 :66
sum : 19 E0 FD 14 28 CE 23 76 C2 C3 43 E4 C0 AD 18 FB :C5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: 03 C0 06 30 0E C8 CD 5D 02 CD 3C 24 21 E0 CE 22 :19
C510: 00 E0 AF 32 02 E0 CD F6 11 CD 14 14 21 FF FF 2B :86
C520: 7D B4 20 FB D8 09 CB 7F CC 6D 24 3A 02 E0 3C 3C :6B
C530: 32 02 E0 D6 0C 20 22 32 02 E0 2A 00 E0 11 10 00 :77
C540: A7 ED 52 22 00 E0 11 00 C0 A7 ED 52 30 0B 3A 10 :24
C550: E0 A7 28 1F 3E F2 C3 CC 24 CD 8C 15 3A 10 E0 A7 :F0
C560: 20 18 3E 06 D3 55 3A 15 E0 3D 32 15 E0 20 23 3E :88
C570: 01 D3 55 AF 32 15 E0 C3 A8 24 3A 15 E0 A7 28 12 :9E
C580: 3D 32 15 E0 20 0C 3A 10 E0 FE 02 20 05 3E 01 32 :50
C590: 10 E0 CD 78 16 CD FB 17 CD 42 1C 3A 10 E0 A7 28 :5B
C5A0: 04 3D 32 1D E0 CD 2C 13 CD 75 12 CD 40 1D CD 45 :0C
C5B0: 16 CD FB 17 3A 10 E0 3D 20 2E DB 08 CB 6F 28 04 :F3
C5C0: CB 77 20 24 3A 47 E0 A7 28 1E 3A 48 E0 D6 04 38 :48
C5D0: 05 32 48 E0 18 0F 3A 47 E0 3D 32 47 E0 3E FF 32 :EC
C5E0: 48 E0 CD 43 10 CD 8C 15 CD 5F 27 CD 83 22 CD 33 :7B
C5F0: 23 C3 24 25 21 2B DC 06 20 0E 0D CD 5D 02 21 00 :E5
sum : FC 3D 2A 21 0D 11 38 28 DC 67 2E 5B 18 94 0C 00 :59

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: A0 2B 00 00 00 7D B4 20 F8 C9 06 05 C5 CD F4 25 :93
C610: 21 1C 0D 11 23 26 0E 07 CD DB 10 CD FE 25 C1 10 :02
C620: EB 18 37 20 20 20 47 41 4D 45 20 4F 56 45 52 00 :10
C630: 43 4F 4E 47 52 41 54 55 4C 41 54 49 4F 4E 53 00 :7D
C640: 28 43 29 20 48 55 52 45 20 45 49 4A 49 20 38 36 :BA
C650: 2F 31 2F 31 20 00 00 00 00 00 3A 0F E0 47 3A 57 :E1
C660: 26 B8 30 12 78 32 57 26 3A 0E E0 32 56 26 3A 0D :64
C670: E0 32 55 26 18 2A B8 20 27 3A 0E E0 47 3A 56 26 :CF
C680: B8 30 0C 78 32 56 26 3A 0D E0 32 55 26 18 11 B8 :F3
C690: 20 0E 3A 0D E0 47 3A 55 26 B8 30 04 78 32 55 26 :62
C6A0: CD DA 01 21 00 C0 06 50 0E C8 CD 5D 02 CD E7 01 :96
C6B0: 21 14 CA 11 27 27 0E 06 CD DB 10 21 28 CF 11 57 :AA
C6C0: 27 0E 07 CD DB 10 21 28 CF DB 10 21 57 26 CD 1A :7E
C6D0: 21 14 D9 11 32 27 0E 05 CD DB 10 21 28 DE 11 57 :D2
C6E0: 27 0E 07 CD DB 10 21 28 DE DD 21 0F E0 CD 1A 10 :FF
C6F0: CD 00 06 21 8E EA 11 3D 27 0E 03 CD DB 10 21 FF :9A
sum : 4E 38 3D 84 3F 6A 93 BF 8E 95 8F 00 FF BA 20 A1 :6E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: FF 2B D8 01 CB 7F 28 1E 7D B4 20 F5 21 8E EA 06 :7B
C710: 38 0E 08 CD 5D 02 21 00 80 2B DB 01 CB 7F 28 06 :9A
C720: 7D B4 20 F5 18 CA C9 48 49 47 48 20 53 43 4F 52 :68
C730: 45 00 59 4F 55 52 20 53 43 4F 52 45 00 54 52 59 :2F
C740: 20 41 47 41 49 4E 20 50 55 53 48 20 52 45 54 55 :40
C750: 52 4E 20 48 45 59 00 30 30 30 30 30 30 00 CD :C6
C760: C6 27 CD 18 28 CD 4A 28 CD 51 2E C9 00 00 0E 0A :69
C770: CD 00 06 3E 03 B9 20 03 32 6C 27 21 00 E1 06 0A :97
C780: C5 E5 CD 92 27 E1 01 08 00 09 C1 10 F3 AF 32 6C :34
C790: 27 C9 7E A7 C8 3C C8 3D 4F E6 0F FE 0C D0 FE 04 :3E
C7A0: D8 FE 07 38 09 FE 09 28 05 3A 6C 27 A7 C8 79 F6 :FD
C7B0: 40 77 23 23 46 3A 12 E0 B8 38 04 06 08 18 02 06 :91
C7C0: 00 2B 70 C9 00 00 CD EC 27 3A 60 E1 A7 C8 3A C4 :2C
C7D0: 27 4F E6 0F C8 FE 01 CA 8F 2B FE 0E D2 AB 2C FE :69
C7E0: 0C D2 79 2D E6 01 CA F4 29 C3 A3 2A 3A 02 E0 FE :FC
C7F0: 0A C0 21 00 D0 06 0A 0E 00 5E 23 56 23 E5 2A 00 :E2
sum : 3F A2 FB 90 0A 24 42 69 F8 9C C6 3F 45 B3 36 19 :25

```



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: E0 A7 ED 52 E1 28 04 0C 10 EF C9 21 F0 D0 06 00 :8E
C810: 09 3E 05 32 60 E1 7E 32 C4 27 C9 21 00 E1 06 0A :35
C820: C5 E5 CD 2E 28 E1 01 08 00 09 C1 10 F3 C9 7E A7 :72
C830: C8 3C CA BF 28 3D 4F E6 0F FE 0C D2 62 32 FE 04 :A8
C840: D2 37 2F 3D CA F3 30 C3 80 30 0E 46 CD D0 06 3A :06
C850: 08 E0 C6 0A B9 D8 21 00 E1 06 0A C5 E5 CD 69 28 :66
C860: E1 01 08 00 09 C1 10 F3 C9 7E A7 C8 3C C8 3D E6 :94
C870: 20 C8 0E 07 CD D0 06 3E 05 B9 C0 7E E6 0F FE 0A :D7
C880: 30 16 ED 5F E6 01 C8 23 23 46 23 4E 3E 05 B9 D0 :0A
C890: 3E 23 B9 D8 0D C3 E2 1F 23 7E F5 23 46 23 4E 3E :71
C8A0: 05 B9 30 19 3E 23 B9 38 14 F1 A7 20 06 04 04 0C :3F
C8B0: 0C 18 06 FE 08 28 02 0C 0C 0D C3 3E 20 F1 C9 CD :27
C8C0: 04 2E A7 20 03 36 00 C9 30 20 05 21 D0 A1 18 16 :10
C8D0: 3D 20 05 21 80 A1 18 0E 3D 20 05 21 30 A1 18 06 :3C
C8E0: CD 06 01 21 D0 A1 C3 A0 05 E5 23 23 73 23 72 E1 :E2
C8F0: C5 D5 E5 CD 0C 29 E1 D1 C1 C9 E5 23 70 23 73 23 :EE
sum : A6 19 02 3C 82 33 5A EE B8 3A 72 CC A6 C5 1B 0E :8E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: 72 E1 C5 D5 E5 CD 99 29 E1 D1 C1 C9 7E E6 0F FE :0E
C910: 04 DA 69 29 FE 0A 38 05 11 00 B0 18 1C FE 07 38 :E7
C920: 0C 28 05 11 E0 B1 18 11 11 C0 B3 18 0C FE 04 28 :D6
C930: 05 11 A0 B5 18 03 11 80 B7 23 7E A7 20 05 01 40 :7C
C940: 01 18 07 FE 08 20 08 01 A0 00 E5 E8 09 E8 E1 23 :87
C950: 23 23 7E A7 20 0C 3C 77 01 50 00 E5 E8 09 E8 E1 :40
C960: 18 02 AF 77 2B 2B C3 DF 29 7E FE 02 28 05 11 60 :7D
C970: B9 18 03 11 A0 BA 23 7E A7 20 05 01 A0 00 18 10 :75
C980: FE 08 20 05 01 50 00 18 07 FE 0C 28 08 01 F0 00 :C6
C990: E5 E8 09 E8 E1 23 C3 DF 29 7E E6 0F FE 0A 30 05 :43
C9A0: 11 E0 B8 18 0C E6 01 28 05 11 20 BD 18 03 11 60 :5E
C9B0: BE 23 23 23 23 7E A7 20 07 3C 77 01 50 00 18 1A :CC
C9C0: FE 01 20 07 3C 77 01 A0 00 18 0F FE 02 20 07 3C :04
C9D0: 77 01 F0 00 18 04 AF 77 18 03 E8 09 E8 2B 2B 05 :CF
C9E0: 5E 23 56 23 23 23 E5 E8 CD 9A 06 E8 E1 73 23 72 :51
C9F0: E1 C3 A0 05 21 00 E1 06 0A C5 E5 3A 60 E1 A7 28 :4F
sum : E2 27 17 4B 77 11 05 DB 56 E5 F8 94 1E 8D 55 3C :D6

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CA00: 03 CD 0D 2A E1 01 08 00 09 C1 10 ED C9 7E A7 C0 :66
CA10: C5 0E 0F CD D0 06 79 C1 FE 04 38 18 FE 08 38 40 :8F
CA20: 3A 60 E1 C8 27 C8 27 C6 10 5F 16 00 CD 16 08 D0 :68
CA30: 06 0C 18 56 3A 60 E1 C8 27 C8 27 C6 04 1E 00 57 :1E
CA40: CD 16 08 30 DB 3A 12 E0 FE 12 30 10 3A 13 E0 C6 :68
CA50: 03 BA 38 08 3A 13 E0 D6 03 BA 38 C4 06 00 18 2A :01
CA60: 3A 60 E1 C8 27 C8 27 C6 04 1E 2D 57 CD 16 08 30 :E9
CA70: AF 3A 12 E0 FE 1C 38 10 3A 13 E0 C6 03 BA 38 08 :2D
CA80: 3A 13 E0 D6 03 BA 38 98 06 08 E5 71 23 70 23 73 :1D
CA90: 23 72 23 36 00 23 36 00 3A 60 E1 3D 32 60 E1 E1 :53
CAA0: C3 0C 29 3A 60 E1 A7 C8 3A 62 E1 A7 28 05 30 32 :A2
CAB0: 62 E1 C9 3A 60 E1 FE 05 CC 23 2B 3E 04 32 62 E1 :5B
CAC0: 32 64 E1 21 00 E1 06 0A C5 E5 3A 60 E1 A7 28 09 :86
CAD0: 3A 64 E1 A7 28 03 CD E2 2A E1 01 08 00 09 C1 10 :EE
CAE0: E7 C9 7E A7 C0 3A 66 E1 5F 3A 68 E1 57 CD 16 08 :3D
CAF0: 38 11 C5 CD 23 2B 3A 66 E1 5F 3A 68 E1 57 CD 16 :C6
sum : CE C5 45 B7 1A 4E 60 76 F2 38 A9 00 42 78 94 F0 :DE

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CB00: 08 C1 D0 71 E5 23 3A 6A E1 77 23 73 23 72 23 36 :95
CB10: 00 23 36 00 3A 60 E1 3D 32 60 E1 AF 32 64 E1 E1 :8B
CB20: C3 0C 29 C5 0E 0F CD D0 06 79 C1 FE 04 38 15 FE :04
CB30: 08 38 22 C5 0E 1A CD D0 06 79 C6 0A 5F 16 00 C1 :71
CB40: 06 0C 18 20 05 0E 05 CD D0 06 79 C6 04 1E 00 57 :7D
CB50: C1 06 00 18 0F C5 0E 05 CD D0 06 79 C6 04 1E 2D :F7
CB60: 57 C1 06 08 7B 32 66 E1 7A 32 68 E1 78 32 6A E1 :04
CB70: C9 3A 60 E1 A7 C8 FE 02 20 08 3A 0B E0 FE 05 D0 :D6
CB80: AF 32 60 E1 C9 3D C0 3A 08 E0 FE 0A 0D 18 F1 3A :28
CB90: 60 E1 A7 C8 3A 62 E1 A7 28 05 3D 32 62 E1 C9 3E :8A
CBA0: 04 32 62 E1 32 64 E1 21 00 E1 06 0A C5 E5 CD 71 :EA
CBB0: 2B 3A 60 E1 A7 28 09 3A 64 E1 A7 28 03 CD C9 2B :90
CBC0: E1 01 08 00 09 C1 10 E4 C9 7E A7 C0 CD F0 2B D0 :0E
CBD0: 71 E5 23 3A 6A E1 77 23 73 23 72 23 36 00 23 36 :52
CBE0: 00 3A 60 E1 3D 32 60 E1 AF 32 64 E1 E1 C3 99 29 :87
CBF0: C5 0E 0F CD D0 06 79 C1 FE 04 DA 48 2C FE 08 DA :EF
sum : 12 E2 32 6F 8D 7E 17 E1 D6 5A EB CF E4 D2 E5 28 :45

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CC00: 53 2C ED 5F E6 01 28 1E C5 E5 1E 00 16 00 2A 00 :00
CC10: E0 01 10 00 09 06 00 CD 7A 2C 38 07 1C 1C 1C 23 :36
CC20: 10 F5 A7 E1 C1 C9 C5 E5 16 00 1E 2D 2A 00 E0 7D :A9
CC30: C6 0F 6F 01 10 00 09 06 00 CD 7A 2C 38 07 10 10 :50
CC40: 1D 2B 10 F5 A7 E1 C1 C9 16 00 1E 00 C5 E5 2A 00 :67
CC50: E0 18 00 16 00 1E 2D C5 E5 2A 00 E0 7D C6 0F 6F :D8
CC60: 01 10 00 09 06 0C CD 7A 2C 38 0C 14 14 14 D5 11 :05
CC70: 10 00 19 01 10 F0 A7 E1 C1 C9 7E FE 08 28 18 FE :CE
CC80: 40 28 1B 38 24 FE 45 38 0E 28 13 FE 46 28 08 FE :15
CC90: 49 38 16 FE 48 30 12 3E 0C 32 6A E1 37 C9 7B A7 :0B
CCA0: 28 04 3E 08 18 F3 AF 18 F0 AF C9 79 E6 0F FE 0E :26
CCB0: DA 79 2D 3A 60 E1 A7 C8 3A 62 E1 A7 28 05 3D 32 :2A
CCC0: 62 E1 C9 3E 04 32 62 E1 32 64 E1 21 00 E1 06 0A :4C
CCD0: C5 E5 CD 71 2B 3A 60 E1 A7 28 09 3A 64 E1 A7 28 :84
CCE0: 03 CD ED 2C E1 01 08 00 09 C1 10 E4 C9 7E A7 C0 :3F
CCF0: CD FA 2C CD 16 08 D0 C3 D0 2B C5 0E 04 CD D0 06 :E9
sum : 99 EE 94 46 8A 45 AC 9A 40 EC 7C 9E AE 1C 4B 18 :E9

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CD00: 0D 28 0B 0E 28 CD D0 06 0C 59 16 00 18 31 ED 5F :2C
CD10: E6 01 28 11 3E 01 32 78 2D 1E 00 0E 03 CD 00 06 :08
CD20: 3E 0D 81 18 0F 1E 2D 3E 05 32 78 2D 0E 03 CD 00 :06
CD30: 06 3E 09 81 F5 0E 08 CD D0 06 0C 51 F1 18 34 7B :91
CD40: FE 1F 38 0F 3E 04 32 78 2D 0E 03 CD D0 06 3E 0A :79
CD50: 81 18 20 FE 0F 38 0F 3E 03 32 78 2D 0E 05 CD D0 :D5
CD60: 06 3E 0A 81 18 0D 3E 02 32 78 2D 0E 03 CD D0 06 :8F
CD70: 3E 0C 81 32 6A E1 C1 C9 00 3A 60 E1 A7 C8 3A 62 :58
CD80: E1 A7 28 05 3D 32 62 E1 C9 3A 60 E1 FE 05 CC B8 :32
CD90: 2D 3E 04 32 62 E1 32 64 E1 21 00 E1 06 0A C5 E5 :17
CDA0: 3A 60 E1 A7 28 09 3A 64 E1 A7 28 03 CD D2 2D E1 :51
CDB0: 01 08 00 09 C1 10 E7 C9 CD FA 2C ED 5F E6 01 28 :E1
CDC0: 08 3A 78 2D C6 05 32 78 2D 7B 32 66 E1 7A 32 68 :91
CDD0: E1 C9 7E A7 C0 3A 66 E1 5F 3A 68 E1 57 C3 D0 2B :07
CDE0: 23 23 73 23 72 23 23 23 E5 EB CD 9A 06 EB E1 73 :33
CDF0: 23 72 C9 23 23 73 23 72 23 E5 EB CD 9A 06 EB E1 :D8
sum : 72 DA DF 79 DF 25 0A 6A 5C 22 A8 D5 AA AE 60 7F :4E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CE00: 73 23 72 C9 E5 C5 23 23 5E 23 56 23 23 7E 3D 77 :10
CE10: F5 23 5E 23 56 D5 EB CD 8B 02 D1 F1 C1 E1 C9 E5 :18
CE20: C5 23 23 5E 23 56 23 23 7E 3C 77 3D 18 14 E5 C5 :6C
CE30: 23 7E 23 5E 23 56 23 23 18 08 E5 C5 23 7E 23 5E :CD
CE40: 23 56 D5 F5 23 5E 23 56 EB CD 8B 02 F1 D1 C1 E1 :E6
CE50: C9 21 80 E1 06 02 C5 E5 CD 64 2E E1 01 08 00 09 :4F
CE60: C1 10 F3 C9 7E A7 C8 CD 3A 2E A7 20 16 1C 1C 1C :E0
CE70: 3E 2E B8 38 65 CD D0 2E 38 60 CD F3 2D 21 30 AA :1C
CE80: C3 A0 05 FE 04 20 16 15 15 15 3E 02 BA 30 4B CD :21
CE90: DD 2E 38 39 CD F3 2D 21 90 A9 C3 A0 05 FE 08 20 :51
CEA0: 16 1D 10 1D 3E 2F B8 38 31 CD DD 2E 38 1F CD F3 :ED
CEB0: 2D 21 E0 A9 C3 A0 05 14 14 14 3E 2D BA 38 1B CD :C0
CEC0: DD 2E 38 09 CD F3 2D 21 40 A9 C3 A0 05 3E 02 32 :1D
CED0: 10 E0 3E 20 32 15 E0 CD 68 01 36 00 C9 3A 10 E0 :D4
CEE0: FE 02 28 05 3D 28 02 A7 C9 E5 3A 12 E0 93 28 10 :E0
CEF0: 3C 28 0D 3C 28 0A 3D 3D 3D 28 05 3D 28 02 18 14 :56
sum : 45 E0 FE E6 C3 36 30 C0 41 7E 04 F8 DB 99 A8 12 :DB

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: 3A 13 E0 92 28 11 3C 28 0E 3C 28 08 3D 3D 3D 28 :88
CF10: 06 3D 28 03 E1 A7 C9 E1 37 C9 3A 10 E0 A7 C8 36 :6F
CF20: FF 23 23 23 23 23 23 23 3A 46 E0 A7 CA 63 16 30 :70
CF30: C0 32 46 E0 C3 8B 08 79 E6 40 28 0F 3E 38 32 87 :73
CF40: 2F 3E 18 32 E1 2F 32 33 30 18 0D 3E 30 32 87 2F :D7
CF50: 3E 38 32 E1 2F 32 33 30 CD 2E 2E A7 CA 2E 30 FE :43
CF60: 08 CA CD 2F 79 E6 40 20 07 79 E6 0F FE 07 38 25 :73
CF70: 3A 10 E0 A7 28 1F 3A 13 E0 92 28 07 3D 28 04 3C :AB
CF80: 3C 20 12 3A 12 E0 93 30 06 23 36 08 2B 18 3E 23 :68
CF90: 36 00 2B 18 38 7E E6 0F FE 05 28 0D FE 08 28 09 :93
CFA0: 14 3E 30 BA 30 0F 36 00 C9 C5 0E 32 CD 00 06 00 :2F
CFB0: C1 28 08 18 EB CD 16 08 38 09 15 ED 5F E6 01 28 :93
CFC0: C8 18 CC CD E9 28 CD D0 2E D0 C3 1A 2F CD 16 08 :2C
CFD0: 38 F1 36 FF 23 23 23 23 23 36 05 C9 79 E6 0F FE :7D
CFE0: 07 38 19 3A 10 E0 A7 28 13 3A 12 E0 93 28 07 3D :8F
CFF0: 28 04 3C 3C 20 06 23 36 0C 2B 18 28 7E E6 0F FE :0B
sum : 24 C0 43 E7 41 37 A1 C5 BE 3D 26 EB 68 A5 E8 55 :42

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D000: 05 28 0D FE 08 28 09 1D 3E 2E B8 30 0F 36 00 C9 :F3
D010: C5 0E 32 CD D0 06 00 C1 28 DC 18 EB CD 16 08 38 :A3
D020: 03 1C 18 D2 CD E9 28 CD D0 2E D0 C3 1A 2F 79 E6 :FA
D030: 0F FE 07 38 19 3A 10 E0 A7 28 13 3A 12 E0 93 28 :58
D040: 07 3D 28 04 3C 3C 20 06 23 36 0C 2B 18 28 7E :E6
D050: 0F FE 05 28 0D FE 08 28 09 1C 3E 2D B8 30 0F 36 :35
D060: 00 C9 C5 0E 32 CD D0 06 00 C1 28 DC 18 EB CD 16 :29
D070: 0B 38 03 1D 18 D2 CD E9 28 CD D0 2E D0 C3 1A 2F :DF
D080: CD 2E 2E A7 28 0E FE 08 28 2B 14 14 3E 30 BA 30 :1F
D090: 03 36 00 C9 CD 16 08 38 12 15 15 ED 5F E6 01 28 :BF
D0A0: 06 23 36 08 2B 18 04 23 36 00 2B CD E9 28 CD D0 :8A
D0B0: 2E D0 C3 1A 2F 1D 1D 3E 2E B8 30 03 36 00 C9 CD :6A
D0C0: 16 08 38 06 1C 1C 23 36 0C 2B CD E9 28 CD D0 2E :DD
D0D0: D0 C3 1A 2F 1C 1C 3E 2D B8 30 03 36 00 C9 CD 16 :4F
D0E0: 08 38 06 1D 1D 23 36 0C 2B CD E9 28 CD D0 2E D0 :99
D0F0: C3 1A 2F CD 2E 2E 47 C5 D5 E5 1C EB CD 41 09 E1 :FA
sum : B5 03 01 DD 23 4C 1B 7D B0 48 5E 7D 41 53 BD 67 :28

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D100: D1 C1 FE 08 28 3F FE 48 D2 31 32 06 40 DA 31 32 :D0
D110: 28 25 3D 28 3F 3D 28 2D 3D CA 7A 31 3D CA 9A 31 :07
D120: 3D CA A6 31 3D CA C4 31 3D CA D0 31 3D CA E8 31 :02
D130: 3D CA 01 32 C3 19 32 78 A7 20 04 1C C3 48 32 1D :04
D140: 06 08 C3 48 32 78 FE 04 20 04 15 C3 48 32 14 06 :5B
D150: 0C C3 48 32 3A 12 E0 B8 28 0E 38 06 06 00 1C C3 :8C
D160: 48 32 06 08 1D C3 48 32 3A 13 E0 BA 38 06 06 0C :1F
D170: 14 C3 48 32 06 04 15 C3 48 32 3A 12 E0 BA 38 06 :CA
D180: 38 06 06 00 1C C3 48 32 3A 13 E0 BA 38 06 06 0C :D7
D190: 14 C3 48 32 06 04 15 C3 48 32 3A 12 E0 B8 30 E8 :B2
D1A0: 06 08 1D C3 48 32 3A 13 E0 BA 30 06 06 04 15 C3 :6A
D1B0: 48 32 3A 12 E0 B8 38 06 06 00 1C C3 48 32 06 08 :12
D1C0: 1D C3 48 32 3A 13 E0 BA 38 E8 06 0C 14 C3 48 32 :CA
D1D0: 78 A7 28 0E FE 04 28 04 FE 08 20 06 15 06 04 C3 :91
D1E0: 48 32 06 00 1C C3 48 32 78 FE 08 28 0E FE 04 28 :BD
D1F0: 04 FE 0C 28 06 15 06 04 C3 48 32 06 08 1D C3 48 :D4
sum : 65 D7 6E B9 9D 53 85 D7 9C 74 AD BE 8E 87 AA B5 :9E

```



D200:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D210:	32	78	FE	08	28	0D	FE	0C	28	03	A7	20	06	14	06	0C	:0D
D220:	C3	4B	32	06	08	1D	C3	4B	32	78	A7	28	0E	FE	0C	28	:32
D230:	04	FE	08	20	06	14	06	0C	C3	4B	32	06	00	1C	C3	4B	:C6
D240:	32	14	C5	05	E5	1C	E8	CD	41	09	E1	D1	C1	FE	08	CA	:26
D250:	45	31	FE	40	38	05	FE	48	DA	0B	31	3E	2D	B8	38	05	:B3
D260:	3E	2F	BA	30	03	36	00	C9	CD	FA	28	CD	DD	2E	D0	C3	:B3
D270:	1A	2F	79	E6	0F	FE	0E	DA	63	35	CD	2E	2E	47	C5	05	:3F
D280:	E5	EB	CD	41	09	E1	D1	C1	FE	0A	CA	D6	33	FE	08	CA	:08
D290:	E7	33	FE	10	38	1C	FE	26	DA	1F	33	FE	50	38	13	FE	:63
D2A0:	68	DA	77	33	FE	80	38	0A	FE	98	DA	8B	33	FE	B2	CA	:54
D2B0:	04	34	CD	A8	32	C3	4B	32	78	FE	04	38	0B	FE	08	38	:1A
D2C0:	20	FE	0C	38	36	C3	05	33	A7	20	03	1C	1C	C9	3D	:8B	
D2D0:	04	1C	1C	15	C9	3D	20	05	1C	1C	15	15	C9	1C	15	:ED	
D2E0:	C9	D6	04	20	03	15	15	C9	3D	20	04	1D	15	15	C9	:67	
D2F0:	20	05	1D	1D	15	15	C9	1D	1D	15	C9	D6	08	20	03	:88	
sum :	2A	4E	C3	2F	F1	1A	30	73	9C	76	67	18	ED	C5	B4	53	:62
D300:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D310:	C9	1D	14	14	C9	D6	0C	20	03	14	14	C9	1C	1C	14	C9	:4A
D320:	14	14	C9	3D	20	05	1C	1C	14	14	C9	1C	1C	14	C9	D6	:67
D330:	10	CA	F8	33	3D	CA	04	34	3D	CA	10	34	3D	CA	30	34	:FA
D340:	3D	CA	50	34	3D	CA	70	34	3D	CA	10	34	3D	CA	30	34	:EC
D350:	3D	CA	50	34	3D	CA	70	34	3D	CA	50	34	3D	CA	70	34	:6C
D360:	3D	CA	04	34	3D	CA	04	34	3D	CA	50	34	3D	CA	70	34	:B4
D370:	3D	CA	50	34	3D	CA	70	34	3D	CA	50	34	3D	CA	70	34	:6C
D380:	3D	CA	D6	33	C3	E7	33	E6	07	FE	02	DA	F8	33	FE	04	:E1
D390:	DA	04	34	FE	06	DA	D6	33	C3	E7	33	FE	8F	CA	A2	32	:01
D3A0:	38	08	FE	94	DA	D6	33	C3	E7	33	D6	80	CA	10	34	3D	:33
D3B0:	CA	30	34	3D	CA	F8	33	3D	CA	F8	33	3D	CA	10	34	3D	:1A
D3C0:	CA	30	34	3D	CA	D6	33	3D	CA	E7	33	3D	CA	10	34	3D	:E7
D3D0:	CA	30	34	3D	CA	10	34	3D	CA	30	34	3D	CA	D6	33	3D	:31
D3E0:	CA	E7	33	C3	04	34	78	FE	05	DA	A2	32	FE	0C	D2	A2	:86
D3F0:	32	CD	90	34	C3	A2	32	78	FE	04	DA	F2	33	FE	0D	DA	:88
sum :	2C	6F	FD	57	16	DB	A2	7B	D2	1D	17	F6	0C	65	98	92	:94
D400:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D410:	34	C3	A2	32	78	FE	08	D2	A2	32	CD	F6	34	C3	A2	32	:7D
D420:	78	FE	05	DA	A2	32	FE	08	30	06	CD	90	34	C3	A2	32	:8D
D430:	FE	0D	38	06	CD	F6	34	C3	A2	32	CD	5C	35	C3	A2	32	:CC
D440:	78	FE	04	30	06	CD	90	34	C3	A2	32	FE	08	DA	A2	32	:8C
D450:	FE	0D	38	06	CD	5C	35	C3	A2	32	CD	F6	34	C3	A2	32	:CC
D460:	78	FE	04	30	06	CD	F6	34	C3	A2	32	FE	09	30	06	CD	:48
D470:	5C	35	C3	A2	32	FE	0C	D2	A2	32	CD	90	34	C3	A2	32	:00
D480:	78	FE	05	30	06	CD	5C	35	C3	A2	32	FE	08	30	06	CD	:AF
D490:	F6	34	C3	A2	32	FE	0D	DA	A2	32	CD	90	34	C3	A2	32	:A2
D4A0:	FE	04	38	08	FE	08	38	1C	FE	0C	38	2E	C3	E0	34	A7	:8D
D4B0:	20	03	06	08	C9	3D	20	03	06	07	C9	3D	20	03	06	06	:9C
D4C0:	C9	06	05	C9	D6	04	20	03	06	05	C9	3D	20	03	06	03	:D7
D4D0:	C9	3D	20	03	06	02	C9	06	01	C9	D6	08	20	03	06	00	:D1
D4E0:	C9	3D	20	03	06	0F	C9	3D	20	03	06	0E	C9	06	00	C9	:20
D4F0:	D6	0C	20	03	06	0D	C9	3D	20	03	06	0B	C9	3D	20	03	:7B
sum :	B7	DB	16	D7	E2	15	3B	4F	26	D8	0E	C3	3F	14	EB	80	:8D
D500:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D510:	38	2E	C3	46	35	A7	20	03	06	01	C9	3D	20	03	06	0F	:B3
D520:	C9	3D	20	03	06	0E	C9	06	0D	C9	D6	04	20	03	06	0C	:F1
D530:	C9	3D	20	03	06	08	C9	3D	20	03	06	0A	C9	06	09	C9	:14
D540:	D6	08	20	03	06	09	C9	3D	20	03	06	07	C9	3D	20	03	:6F
D550:	06	06	C9	06	05	C9	D6	0C	20	03	06	04	C9	3D	20	03	:E1
D560:	06	03	C9	3D	20	03	06	02	C9	06	01	C9	78	C6	08	E6	:FF
D570:	0F	47	C9	3A	78	2D	A7	28	04	FE	0B	38	06	36	00	CD	:1B
D580:	1F	2E	C9	CD	1F	2E	C5	D5	E5	4F	06	00	3A	78	2D	3D	:20
D590:	C8	27	5F	16	00	21	E5	35	19	5E	23	56	EB	09	7E	47	:4B
D5A0:	E6	0F	4F	78	0F	0F	0F	0F	E6	0F	47	E1	D1	E6	08	20	:F4
D5B0:	1C	78	E6	07	82	57	79	E6	08	20	07	79	E6	07	83	5F	:30
D5C0:	18	28	79	E6	07	4F	78	91	38	27	5F	18	20	78	E6	07	:5F
D5D0:	47	7A	90	38	1C	57	79	E6	08	20	07	79	E6	07	83	5F	:D2
D5E0:	18	08	79	E6	07	4F	78	91	38	07	5F	18	00	C1	C3	4B	:69
D5F0:	32	C1	36	00	C9	00	D1	40	D1	80	D1	C0	D1	00	D2	40	:C8
sum :	22	CD	65	F2	59	6C	43	40	48	D0	D0	70	06	AB	BE	CE	:20
リスト7-4 地表キャラクタ "DATA 1"																	
C000:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C010:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C020:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
C030:	88	88	88	22	22	22	88	88	88	22	22	88	88	88	22	88	:B6
C040:	22	22	88	88	88	22	22	22	AA	AA	AA	55	55	55	AA	AA	:93
C050:	AA	55	55	55	AA	AA	AA	55	55	55	AA	AA	55	55	55	55	:A3
C060:	AA	AA	AA	55	55	55	AA	AA	AA	55	55	55	FF	FF	FF	FF	:F6
C070:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C080:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:F0
C090:	FF	FF	FF	FF	00	00	FF	FF	FF	00	00	FF	FF	FF	FF	00	:F7
C0A0:	00	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	FF	FF	:F8
C0B0:	FF	00	00	00	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	:8F
C0C0:	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	:C0
C0D0:	CC	CC	CC	CC	CC	CC	CC	CC	FF	FF	FF	CC	CC	CC	FF	FF	:BF
C0E0:	FF	CC	CC	CC	FF	FF	FF	CC	CC	CC	FF	FF	FF	CC	CC	CC	:25
C0F0:	FF	FF	FF	CC	CC	CC	FF	FF	FF	CC	CC	CC	88	BC	88	22	:B0
sum :	90	09	6E	81	5D	F8	E5	F7	B1	C4	B2	2B	F6	3C	8F	C4	:90
C100:	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C110:	02	22	88	BC	88	22	02	22	88	BC	88	22	02	22	88	BC	:8C
C120:	88	22	02	22	88	BC	88	22	02	22	88	BC	88	22	02	22	:F2
C130:	88	66	88	22	66	22	88	66	88	22	66	22	88	66	88	22	:D8
C140:	66	22	88	66	88	22	66	22	88	66	88	22	66	22	88	66	:B6
C150:	88	22	66	22	88	88	88	22	22	22	88	88	22	22	22	22	:2E
C160:	8F	89	F0	38	E7	1C	60	3C	06	63	BD	C6	63	BD	C6	38	:E9
C170:	E7	1C	8F	89	F0	22	22	22	AA	AA	88	D5	55	22	AA	AA	:ED
C180:	88	D5	55	22	AA	AA	88	D5	55	22	AA	AA	88	D5	55	22	:24
C190:	AA	AA	88	D5	55	22	AA	AA	88	D5	55	22	AA	AA	88	D5	:4F
C1A0:	55	55	88	AA	AB	22	55	55	88	AA	AB	22	55	55	88	AA	:2E
C1B0:	AB	22	55	55	88	AA	AB										



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: 08 10 00 C1 00 00 C8 10 00 C1 00 00 88 88 88 22 :EC
C610: 22 22 FF FF FF 00 00 03 88 88 13 00 00 83 88 80 :F2
C620: 13 00 00 83 00 08 13 00 00 83 00 08 13 00 00 83 :D2
C630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C640: 00 00 00 00 00 22 22 22 22 22 22 22 22 22 22 :CC
C650: 00 FF FF FF 88 88 88 22 22 22 FF FF FF 00 00 00 :F8
C660: 88 88 88 00 00 00 88 88 88 00 00 00 00 00 00 :30
C670: 00 00 00 00 00 00 00 00 00 08 13 00 00 83 00 08 :A6
C680: 13 00 00 00 00 08 88 00 00 00 00 00 88 00 00 :28
C690: 00 00 00 00 00 00 00 00 00 00 00 00 C8 10 00 C1 :99
C6A0: 00 00 C8 10 00 01 00 00 88 10 00 00 00 00 88 80 :79
C6B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C6C0: 88 88 88 22 22 22 88 88 88 22 22 22 88 88 CC :60
C6D0: CC CC CC CC CC 22 22 22 88 88 88 22 22 22 88 88 :70
C6E0: 88 22 22 22 88 BC 88 22 02 22 88 BC 88 22 02 22 :12
C6F0: 88 BC 88 CC CC CC CC F0 CC 22 02 22 88 BC 88 22 :EC
sum : FC EB 4C 2E C9 87 93 98 98 F4 59 4B C6 48 32 06 :55

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: 02 22 88 BC 88 22 02 22 88 BC 88 22 02 22 88 BC :8C
C710: 88 22 02 22 88 BC 88 22 02 22 88 BC 88 22 02 22 :F2
C720: 88 BC 88 22 02 22 88 BC 88 22 02 22 88 BC 88 22 :12
C730: 02 22 88 BC 88 22 02 22 88 BC 88 22 0C CC 88 80 :34
C740: CC 22 02 22 88 BC 88 22 02 22 88 BC 88 22 02 22 :36
C750: 88 BC 88 22 02 22 88 BC 88 22 02 22 88 BC 88 CC :8C
C760: CE 22 CC F0 88 22 02 22 88 BC 88 22 02 22 88 BC :D0
C770: 88 22 02 22 88 BC 88 22 02 22 88 BC 88 22 02 22 :F2
C780: 88 BC 88 CC CC CC F0 CC 22 22 22 88 88 88 22 :D8
C790: 22 22 88 88 88 22 22 22 88 88 88 22 22 88 88 :50
C7A0: 88 22 22 22 88 88 88 CC CC CC CC F0 CC 22 02 22 :88
C7B0: 88 BC 88 22 02 22 88 BC 88 22 02 22 88 BC 88 22 :12
C7C0: 02 22 88 BC 88 22 02 22 88 BC 88 22 0C CC 88 80 :34
C7D0: CC 22 22 22 88 88 88 22 22 22 88 88 88 22 22 :2E
C7E0: 88 BC 88 22 02 22 88 BC 88 22 02 22 88 BC 88 CC :8C
C7F0: CE 22 CC F0 88 22 22 22 88 88 88 22 22 22 88 88 :A8
sum : 9C 22 AA 9A AC 64 E0 00 10 FE 46 22 F4 42 02 90 :30

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: 88 22 22 22 88 88 88 22 22 22 88 88 88 22 22 :EA
C810: 88 88 88 CC CE 22 CC F0 88 22 02 22 88 BC 88 22 :CC
C820: 02 22 88 BC 88 22 02 22 88 88 88 22 22 22 88 88 :44
C830: 88 22 22 22 88 88 88 22 0E 66 88 80 66 22 02 22 :00
C840: 88 BC 88 22 02 22 88 BC 88 22 02 22 88 88 88 22 :DE
C850: 22 22 88 9C 88 22 3E 22 88 BE 88 22 3E 22 88 BE :08
C860: 88 22 1C 22 22 88 9C 88 22 22 22 88 9C 88 22 1C :06
C870: 88 88 88 22 22 22 BF FF FC 32 22 2E 88 88 8C 32 :38
C880: 22 2E 88 88 8C 32 22 2E 88 88 8C 32 22 2E 88 88 :2C
C890: 8C 3F FF FE 88 88 88 22 22 22 BF FF FC 30 00 0E :BE
C8A0: 80 00 0C 30 00 0E 80 00 0C 30 00 0E 80 00 0C :E0
C8B0: 00 0E 80 00 0C 3F FF FE 88 88 88 22 22 22 88 FF :88
C8C0: 08 23 22 E2 8C 88 80 1A 22 1A 98 88 98 1A 22 1A :57
C8D0: 98 88 98 2E 22 72 88 89 C8 22 7E 22 88 88 88 22 :D2
C8E0: 22 22 88 FE 88 23 81 E2 8E 00 70 18 00 1A 98 00 :A0
C8F0: 18 18 00 1A 98 00 18 2E 00 72 88 81 C8 22 7E 22 :30
sum : 8C D6 BD AC 88 7A 18 56 54 76 B2 2E 76 D4 F8 45 :6C

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: 88 88 88 22 22 22 88 88 88 22 22 22 88 88 FF :93
C910: FF FF 00 00 00 FF FF FF 88 88 88 22 22 22 88 88 :09
C920: 88 22 22 22 88 66 88 22 66 22 88 66 22 66 22 :2E
C930: 88 66 88 FF E7 FF 00 00 00 FF E7 FF 88 66 88 22 :D8
C940: 66 22 88 66 88 22 66 22 88 66 88 22 66 22 88 66 :B6
C950: 88 22 66 22 88 66 88 22 66 22 88 66 88 22 66 :72
C960: 88 66 88 22 66 22 88 66 88 22 66 22 88 66 88 :D8
C970: 66 22 88 66 88 22 66 22 88 66 88 22 67 FF 88 60 :8E
C980: 00 22 67 FF 88 66 88 22 66 22 88 66 88 22 66 :C8
C990: 88 66 88 22 66 22 88 66 88 22 66 22 88 66 FF :85
C9A0: E6 22 00 06 88 FF E6 22 88 66 88 22 66 22 88 :AB
C9B0: 88 22 66 22 88 66 88 22 66 22 88 66 88 22 66 :72
C9C0: 88 66 88 FF E7 FF 00 00 00 FF FF FF 88 88 88 22 :12
C9D0: 22 22 88 88 88 22 22 22 88 88 88 22 22 22 88 88 :50
C9E0: 88 22 22 22 88 88 88 FF FF FF 00 00 00 FF E7 FF :68
C9F0: 88 66 88 22 66 22 88 66 88 22 66 22 88 66 88 :D8
sum : 23 B7 3F 67 EA 0A 9B C8 5F 4F 02 C8 C7 B6 57 49 :6C

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CA00: 66 22 88 66 88 22 66 22 88 66 88 22 67 FF 88 60 :8E
CA10: 00 22 7F FF 88 88 88 22 22 22 88 88 88 22 22 :9C
CA20: 88 66 88 22 66 22 88 66 88 22 66 22 88 66 FF :B5
CA30: E6 22 00 06 88 FF FE 22 88 88 88 22 22 22 88 88 :C3
CA40: 88 22 22 22 88 88 88 22 22 22 88 88 88 22 22 :EA
CA50: 88 88 88 FF FE 22 00 06 88 FF E6 22 88 66 88 22 :E4
CA60: 66 22 88 66 88 22 66 22 88 88 88 22 22 22 88 88 :B6
CA70: 88 22 22 22 88 88 88 22 7F FF 88 60 00 22 67 FF :96
CA80: 88 66 88 22 66 22 88 66 88 22 66 22 88 88 88 :FA
CA90: 22 22 88 8E 88 21 E3 C2 88 80 E8 23 80 E2 89 C3 :9C
CAA0: C8 22 0E 22 88 9C 88 22 00 22 88 9C 88 22 1C :16
CAB0: 88 88 88 22 22 22 88 98 88 22 7E 22 89 FF 88 27 :9F
CAC0: FF E2 88 7E 08 22 7E 22 88 7E 88 22 7E 22 88 7E :07
CAD0: 88 22 22 22 88 88 88 22 22 22 88 7E 88 22 7E :3C
CAE0: 88 7E 88 22 7E 22 88 7E 08 27 FF E2 89 FF 88 22 :98
CAF0: 7E 22 88 98 88 22 22 22 88 88 88 22 22 22 88 80 :B4
sum : 59 90 43 B4 C2 0E 15 FE 40 0F 67 C1 95 65 1E 44 :96

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CB00: 88 22 32 22 88 F0 00 23 FF FA 8F FF F8 23 FF FA :34
CB10: 88 F0 00 22 32 22 88 80 88 22 22 22 88 88 88 :9E
CB20: 22 22 88 80 88 22 26 22 80 07 88 1F FF E2 9F FF :EB
CB30: F8 1F FF E2 80 07 88 22 26 22 88 80 88 22 22 :67
CB40: 88 88 88 22 22 22 88 88 88 22 22 22 88 88 88 :B6
CB50: 22 22 AA CC CC 55 6C CC AA BC CC 55 57 EC AA AA :31
CB60: BF FF FF FF 88 88 88 22 22 22 88 88 88 22 22 :88
CB70: 88 88 88 22 22 22 CC CB AA CC CE 55 CC FC AA CF :6F
CB80: E5 55 FC AA AA FF FF FF FF FF FF FF FF FF :7F
CB90: FF 55 55 3F AA A7 EC 55 3C CC AA EC CC 55 CC CC :D1
CBA0: 88 88 88 22 22 22 88 88 88 22 22 22 FF FF FF :F8
CBB0: FF FF FF FF FF FD 55 55 CF EA AA CC FD 55 CC CF :BE
CBC0: 2A CC CB 95 88 88 88 22 22 22 88 88 88 22 22 :52
CBD0: AA AA 88 D5 55 22 AA AA 88 D5 5F A2 AA FF 88 D3 :DE
CBE0: 80 22 A7 FF 88 CC 00 22 9F FF 88 D0 00 22 BF FF :94
CBF0: 88 B0 00 22 BF FF 88 B0 00 22 9F FF 88 D8 00 22 :92
sum : 62 FD 44 4A F3 96 00 F7 06 00 88 E6 BB 04 45 A9 :8E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CC00: AF FF 88 D6 00 22 AB FF 88 D4 F0 22 AA 9F 88 D5 :EC
CC10: 55 22 AA AA 88 D5 55 22 88 AA AB 22 55 55 88 AA :7A
CC20: AB 22 F9 55 88 FF 2B 22 01 D5 88 FF EB 22 00 35 :8E
CC30: 88 FF FB 22 00 09 88 FF FD 22 00 0D 88 FF F0 :06
CC40: 00 0D 88 FF FB 22 00 19 88 FF F3 22 00 65 88 FF :52
CC50: CB 22 0F 55 88 FA AB 22 55 55 88 AA AB 22 55 55 :F3
CC60: 88 88 88 22 22 22 88 88 88 22 22 22 88 88 88 :B6
CC70: 22 22 AA EA AA 55 7F FF AA BF FF 55 57 FF AA AA :8C
CC80: BF FF FF FF 88 88 88 22 22 22 88 88 88 22 22 :88
CC90: 88 88 88 22 22 22 AA AB AA FF FE 55 FF FC AA FF :F3
CCA0: E5 55 FC AA AA FF FF FF FF FF FF FF FF FF :7F
CCB0: FF 55 55 3F AA A7 FF 55 3F FF AA FF FF 55 D5 55 :F2
CCD0: 88 88 88 22 22 22 88 88 88 22 22 22 FF FF FF :F8
CCE0: FF FF FF FF FF FD 55 55 FF EA AA FF FD 55 FF FF :84
CCF0: 2A 55 57 95 88 88 88 22 22 22 88 88 88 22 22 :67
sum : 32 D2 2D EC 5B AB A4 CE 58 CC A1 B9 AF 09 64 5E :8D

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CD00: FF 22 A7 FE 88 CF FF 22 9F FE 88 DF FF 22 BF FE :20
CD10: 88 BF FF 22 BF FE 88 BF FF 22 9F FE 88 DF FF 22 :B2
CD20: AF FE 88 D7 FF 22 AB FE 88 D4 FF 22 AA 9F 88 D5 :F9
CD30: 55 22 AA AA 88 D5 55 22 88 AA AB 22 55 55 88 AA :7A
CD40: AB 22 F9 55 88 BF 2B 22 7F D5 88 BF EB 22 7F F5 :CB
CD50: 88 BF FB 22 7F F9 88 BF FD 22 7F FD 88 BF FD :24
CD60: 7F FD 88 BF FB 22 7F F9 88 BF F3 22 7F E5 88 BF :5F
CD70: CB 22 7F 55 88 FA AB 22 55 55 88 AA AB 22 55 55 :63
CD80: 88 88 88 22 22 22 88 88 88 22 22 22 88 88 88 :B6
CD90: 22 22 AA C8 88 55 62 22 AA BC 00 55 57 E0 AA AA :5D
CDA0: BF FF FF FF 88 88 88 22 22 22 88 88 88 22 22 :88
CDB0: 88 88 88 22 22 22 88 83 AA 22 06 55 00 3C AA 07 :1D
CDC0: E5 55 FC AA AA FF FF FF FF FF FF FF FF FF :7F
CDD0: FF 55 55 3F AA A7 E0 55 3C 00 AA E0 88 55 C2 22 :F5
CDE0: 88 88 88 22 22 22 88 88 88 22 22 22 FF FF FF :F8
CDF0: FF FF FF FF FD 55 55 07 EA AA 00 3D 55 88 87 :DE
sum : 64 63 64 41 21 7E 1A 7D CF D6 78 FE 4D 4B 6D 66 :28

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CE00: 2A 22 23 95 88 88 88 22 22 22 88 88 88 22 22 :00
CE10: AA AA 88 D5 55 22 AA AA 88 D5 5F A2 AA F0 08 D3 :4F
CE20: 82 22 A6 08 88 CC 22 22 98 08 88 80 22 22 80 08 :DE
CE30: 88 B0 22 22 80 08 88 B0 22 22 90 08 88 08 22 22 :EC
CE40: AC 08 88 D6 02 22 AB 80 88 D4 F0 22 AA 9F 88 D5 :75
CE50: 55 22 AA AA 88 D5 55 22 88 AA AB 22 55 55 88 AA :7A
CE60: AB 22 F9 55 88 BF 2B 22 21 05 88 88 68 22 22 :69
CE70: 88 88 1B 22 22 09 88 88 0D 22 22 0D 88 88 0D :25
CE80: 22 0D 88 88 0B 22 22 19 88 88 33 22 20 65 88 81 :9A
CE90: CB 22 0F 55 88 FA AB 22 55 55 88 AA AB 22 55 55 :F3
CEA0: 88 88 88 22 22 22 AA AA AA 55 55 55 AA AA 55 :4E
CEB0: 55 55 AA AA AA 55 55 55 AA AA AA 55 55 55 AA :F8
CEC0: AA 55 55 55 AA AA 55 55 55 AA AA AA 55 55 55 :A3
CED0: AA AA AA 55 55 55 AA AA AA 55 55 55 AA AA 55 :4D
CEE0: 55 55 AA AA AA 55 55 55 88 88 88 22 22 22 FF :A3
CEF0: FF 00 00 00 FF FF FF 00 00 00 FF FF FF 00 00 :F9
sum : 84 D2 2B 88 50 F3 03 78 5A A4 84 71 0D 51 6A 73 :F5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: FF FF FF 00 00 00 FF FF FF 00 00 00 FF FF FF :F7
CF10: 00 00 FF FF FF 00 00 00 FF FF FF 00 00 00 FF :F8
CF20: FF 00 00 00 FF FF FF 00 00 00 FF FF FF 22 22 :5F
CF30: 88 88 88 22 22 22 CC CC CC CC CC CC CC CC :F6
CF40: CC CC CC CC CC CC CC CC CC CC CC CC CC CC :C0
CF50: CC CC CC CC CC CC CC CC CC CC CC CC CC CC :C0
CF60: CC CC CC CC CC CC CC CC CC CC CC CC CC CC :C0
CF70: CC CC CC CC CC CC CC CC 00 00 00 00 00 00 :60
CF80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CF90: 00 00 00 00 00 00 00 00 00 00 00 00 FF FF :FC
CFA0: FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
CFB0: FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
CFC0: 88 88 88 22 22 22 88 FF 08 23 81 C2 87 08 E8 :91
CFD0: 22 E2 87 08 E8 27 22 E2 87 08 E8 23 81 C2 88 :0A
CFE0: 08 22 22 22 88 88 88 22 22 22 88 BC 88 22 7C :F8
CFF0: 89 FC 88 22 3C 22 88 BC 88 22 3C 22 88 BC 88 :C7
sum : EF 3D 6D BD 1C 42 B2 B8 65 9C 59 F0 43 F6 C1 B8 :1A

```



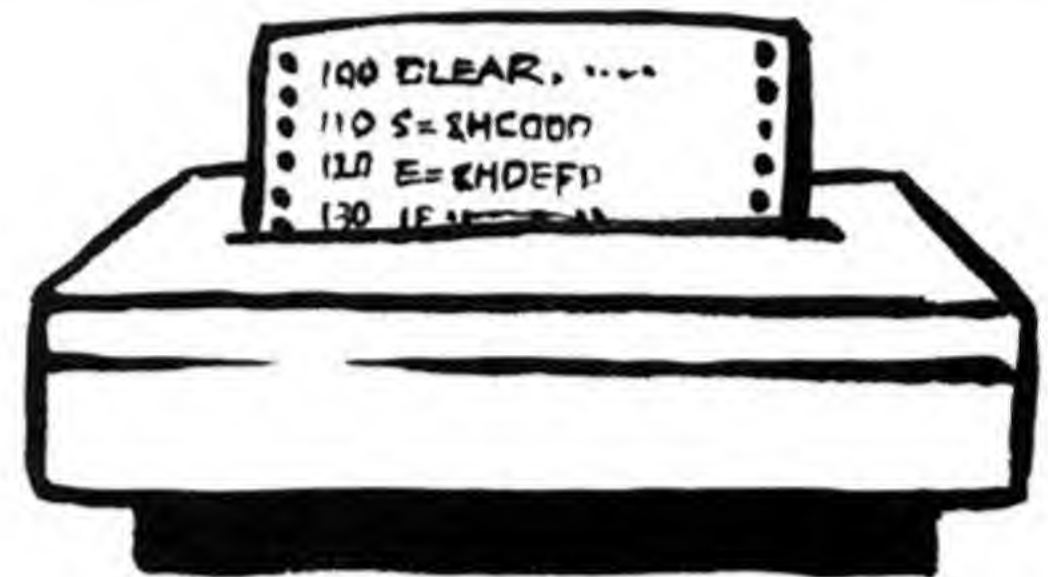
	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D000:	3C	22	88	BC	88	22	22	22	88	88	88	22	22	22	88	7E	:94
D010:	08	23	C3	C2	87	08	E8	27	22	E2	88	83	C8	22	3E	22	:A7
D020:	88	E0	88	23	82	22	87	FF	E8	22	22	22	88	88	88	22	:45
D030:	22	22	88	7E	08	23	C3	C2	87	08	E8	20	03	C2	88	3E	:1C
D040:	08	20	03	C2	87	08	E8	23	C3	C2	88	7E	08	22	22	22	:80
D050:	88	88	88	22	22	22	88	89	C8	22	27	C2	88	9D	C8	22	:F1
D060:	71	C2	89	C9	C8	27	21	C2	87	FF	E8	22	21	C2	88	89	:0B
D070:	C8	22	22	22	88	88	88	22	22	22	87	FF	C8	27	22	22	:E5
D080:	87	08	88	27	7F	22	87	81	C8	20	22	E2	87	08	E8	23	:6D
D090:	81	C2	88	FF	08	22	22	22	88	88	88	22	22	22	88	FF	:8D
D0A0:	08	23	A3	C2	87	08	88	27	7F	22	87	89	C8	27	22	E2	:72
D0B0:	87	08	E8	23	A3	C2	88	FF	08	22	22	22	88	88	88	22	:AE
D0C0:	22	22	89	FF	E8	23	A0	E2	88	89	C8	22	23	A2	88	87	:2B
D0D0:	08	22	2E	22	88	9C	88	22	3C	22	88	BC	88	22	22	22	:08
D0E0:	88	88	88	22	22	22	88	FF	08	23	A1	C2	87	08	E8	23	:AD
D0F0:	A1	C2	89	FF	88	27	A1	E2	87	08	E8	27	A1	E2	89	FF	:C6
sum :	A1	56	F4	3B	5D	5E	77	48	7A	5B	64	BE	BA	BD	9F	E0	:8D

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D100:	88	22	22	22	88	88	88	22	22	22	88	FF	08	23	A1	C2	:01
D110:	87	08	E8	27	20	E2	88	89	E8	21	FE	E2	88	88	E8	23	:88
D120:	81	C2	88	FF	08	22	22	22	88	88	88	22	22	22	70	9F	:45
D130:	FC	F9	FE	7A	FD	83	F8	FE	01	FC	9F	3F	FE	E7	A2	6B	:80
D140:	F9	C9	FF	FE	7B	55	8F	97	FF	F0	75	55	AF	FB	FF	AB	:8F
D150:	31	51	AD	AC	CD	C1	86	18	8F	F2	D3	C1	80	C0	AD	BE	:C7
D160:	D7	A8	30	0D	AA	88	95	AA	22	35	A8	88	55	72	22	D5	:72
D170:	88	88	88	22	22	22	8F	F8	0E	1E	7F	9F	9F	C1	BF	3F	:50
D180:	80	7F	7F	FC	F9	D6	21	E7	FF	88	9F	55	5E	7F	FF	E9	:94
D190:	F1	55	6E	0F	FF	DF	F5	45	8D	55	B3	35	55	18	60	05	:77
D1A0:	C8	4F	F3	03	00	05	E8	55	55	80	05	55	A8	89	55	AC	:E6
D1B0:	22	55	AA	88	95	AB	22	2E	FF	FF	FF	FF	FF	FF	AA	AC	:87
D1C0:	AA	3F	FF	FC	8A	AA	AB	2F	FF	FA	8A	AA	AB	27	FF	F2	:0C
D1D0:	8A	AA	A8	23	FF	E2	8A	AA	A8	22	22	22	00	00	00	00	:22
D1E0:	00	00	00	88	80	05	55	40	02	22	20	17	FF	F0	08	88	:7C
D1F0:	90	5F	FF	FC	22	22	24	7F	FF	FE	88	88	88	22	22	22	:CC
sum :	37	EF	24	D4	79	E7	64	63	D9	C7	C6	C8	5C	FA	AF	49	:C1

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D200:	88	88	88	22	22	22	A8	88	88	52	22	22	AA	88	88	55	:5B
D210:	22	22	AA	A8	88	55	52	22	AA	AA	88	55	55	22	AA	AA	:E3
D220:	A8	D5	55	52	88	88	88	22	22	22	88	88	8A	22	22	25	:25
D230:	88	88	AA	22	22	55	88	8A	AA	22	25	55	88	AA	AA	22	:A9
D240:	55	55	8A	AA	AA	25	55	55	88	88	88	22	22	22	88	88	:65
D250:	88	22	22	22	88	88	88	22	22	22	AA	E8	88	55	D2	22	:4F
D260:	AB	AA	88	57	55	22	AE	AA	A8	FD	55	52	88	88	88	22	:09
D270:	22	22	88	88	88	22	22	22	88	88	88	22	22	22	88	88	:53
D280:	AA	22	25	D5	88	AA	EA	22	55	75	8A	AA	BA	25	55	5F	:95
D290:	AA	AA	88	D5	55	22	AA	AA	88	D5	55	22	AA	AA	88	D5	:01
D2A0:	55	22	AA	AA	88	D5	55	22	AA	A0	88	D4	15	22	82	AA	:A8
D2B0:	A8	D5	55	52	88	AA	AB	22	55	55	88	AA	AB	22	55	55	:76
D2C0:	88	AA	AB	22	55	55	88	AA	AB	22	55	55	88	0A	AB	22	:B1
D2D0:	50	55	8A	AA	83	25	55	55	88	88	88	22	22	22	A8	88	:59
D2E0:	88	52	22	22	AA	88	88	55	22	22	AA	A8	88	55	55	22	:17
D2F0:	AA	AF	88	55	F5	22	BE	AA	88	D5	55	22	88	88	88	22	:43
sum :	DF	0D	78	D2	C7	B4	6E	A7	91	4F	31	5D	43	B3	4C	BE	:34

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D300:	22	22	88	88	8A	22	22	25	88	88	AA	22	22	55	88	8A	:AC
D310:	AA	22	25	55	88	FA	AA	22	57	D5	88	AA	BE	22	55	55	:7C
D320:	FA	AA	AA	FD	55	55	FF	AA	AA	FF	D5	55	FF	FA	AA	FF	:13
D330:	FD	55	FF	FF	AA	FF	FF	D5	FF	FF	FA	FF	FF	FD	FF	FF	:8E
D340:	FF	FF	FF	FF	AA	AA	AF	55	55	5F	AA	AA	FF	55	55	FF	:04
D350:	AA	AF	FF	55	5F	FF	AA	FF	FF	55	FF	FF	AF	FF	FF	5F	:12
D360:	FF	FF	FF	FF	FF	FF	FF	FF	88	88	88	22	22	22	A8	88	:26
D370:	88	52	22	22	AA	88	88	55	22	22	AA	AA	AA	51	55	55	:6A
D380:	AA	AA	AA	15	55	55	AA	AA	AA	FF	FF	FF	88	88	88	22	:72
D390:	22	22	88	88	8A	22	22	25	88	88	AA	22	22	55	AA	AA	:EE
D3A0:	AA	55	55	15	AA	AA	AA	55	55	51	AA	AA	AB	FF	FF	FF	:5E
D3B0:	AA	AA	88	D5	55	22	AB	EA	88	CC	19	22	90	04	88	A0	:08
D3C0:	02	22	A0	02	88	A0	02	22	A0	02	88	D0	05	22	AC	1A	:F9
D3D0:	88	D3	E5	22	88	AA	AB	22	55	55	88	A7	CB	22	58	35	:B4
D3E0:	88	A0	08	22	40	05	88	40	05	22	40	05	88	40	05	22	:8D
D3F0:	20	09	88	98	33	22	57	D5	FF	FF	FF	FF	FF	FF	FF	FF	:C2
sum :	45	AB	9C	B3	24	54	57	D5	8E	D5	97	FD	94	98	98	F3	:91

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
D400:	FF	5F	FF	F5	A8	00	1A	58	00	15	A8	00	1A	5F	FF	F5	:96
D410:	88	88	88	22	22	22	88	88	88	22	22	22	88	88	88	22	:86
D420:	22	22	88	88	88	22	22	22	88	88	88	22	22	22	88	88	:50
D430:	88	22	22	22	88	88	88	22	22	22	88	88	88	22	22	22	:EA
D440:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D450:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D460:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D470:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D480:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D490:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
D4F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
sum :	31	2B	31	C1	DA	CC	4C	24	32	E1	DA	CC	4C	2B	31	C1	:86



## リスト7-5 地上物キャラクタ "DATA 2"

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C000:	00	00	00	00	00	00	AA	AA	88	00	00	00	00	00	00	D5	:81
C010:	55	22	03	F0	00	03	F0	00	AB	FA	88	0F	F0	00	0C	30	:C5
C020:	00	DC	35	22	1F	C8	00	11	48	00	B1	4A	88	3F	84	00	:89
C030:	22	84	00	E2	85	22	3F	84	00	3F	84	00	BF	86	88	3F	:C1
C040:	84	00	3F	84	00	FF	85	22	3F	84	00	22	84	00	A2	86	:7E
C050:	88	1F	C8	00	11	48	00	D1	4D	22	0F	F0	00	0C	30	00	:43
C060:	AC	3A	88	03	E0	00	03	E0	00	D7	F5	22	00	00	00	00	:22
C070:	00	00	00	00	00	00	AA	AA	88	00	00	00	00	00	00	D5	:81
C080:	55	22	03	F0	00	03	F0	00	AB	FA	88	0F	F0	00	0C	30	:C5
C090:	00	DC	35	22	1F	C8	00	11	48	00	B1	4A	88	3F	84	00	:89
COA0:	22	84	00	E2	85	22	3F	84	00	3F	84	00	BF	86	88	3F	:C1
COB0:	84	00	3F	84	00	FF	85	22	3F	84	00	22	84	00	A2	86	:7E
COC0:	88	1F	C8	00	11	48	00	D1	4D	22	0F	F0	00	0C	30	00	:43
COD0:	AC	3A	88	03	E0	00	03	E0	00	D7	F5	22	00	00	00	00	:22
COE0:	00	00	00	00	00	00	AA	AA	88	00	00	00	00	00	00	D5	:81
COF0:	55	22	03	F0	00	03	F0	00	AB	FA	88	0F	F0	00	0C	30	:C5
sum :	B3	D8	91	E6	2A	68	5C	CE	41	66	0A	29	66	A2	E0	99	:1C



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: 22 22 0F FF F0 0C 00 30 8C 00 30 33 FF CC 3F 00 :77
C410: FC 33 00 CE 2C FF 3C 0F FF F0 8C FF 30 17 00 FC :30
C420: 03 FF C0 03 00 C2 2B 00 FC 03 FF C0 83 3C C0 17 :06
C430: 00 FC 03 FF C0 03 3C C2 2B 00 FC 03 FF C0 83 3C C0 :2B
C440: C0 1C FF 3C 0F FF F0 0C FF 32 33 AA CC 3F 00 FC :36
C450: B3 00 CC 0D 55 70 0C 00 30 2C 00 32 00 00 00 00 :EB
C460: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C470: 22 22 0F FF F0 0C 00 30 8C 00 30 33 FF CC 3F FF :76
C480: FC 33 FF CE 2C 00 3C 0F FF F0 8C 00 30 1C 00 3C :76
C490: 0F FF F0 0C FF 32 2C 00 3C 0F FF F0 8C FF 30 1C :78
C4A0: 00 3C 0F FF F0 0C FF 32 2C 00 3C 0F FF F0 8C FF :68
C4B0: 30 1C 00 3C 0F FF F0 0C 00 32 33 FF CC 3F FF FC :FC
C4C0: B3 FF CC 0D 55 70 0C 00 30 2C 00 32 00 00 00 00 :EA
C4D0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C4E0: 22 22 0F FF F0 0C 00 30 8C 00 30 33 FF CC 3F 00 :77
C4F0: FC 33 00 CE 2C FF 3C 0F FF F0 8C FF 30 17 00 FC :30
sum : C2 6C 85 06 CB 03 4E 09 9F 9E D0 66 32 17 BB A1 :C6

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: 03 FF C0 03 00 C2 2B 00 FC 03 FF C0 83 3C C0 17 :06
C510: 00 FC 03 FF C0 03 3C C2 2B 00 FC 03 FF C0 83 3C C0 :2B
C520: C0 1C FF 3C 0F FF F0 0C FF 32 33 AA CC 3F 00 FC :36
C530: B3 00 CC 0D 55 70 0C 00 30 2C 00 32 00 00 00 00 :EB
C540: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C550: 22 22 00 00 00 00 3F FF 00 80 00 08 00 00 0C 30 1C :62
C560: 0C 00 00 2E 00 00 3C 3F FF 3C 80 00 3C 1E 18 FC :DE
C570: 1E 18 FC 1E 18 FE 3F 7E FC 3F 7E F4 8F 7E F4 3F :40
C580: 7E FC 3D 76 EC 3D 76 EE 1E 18 FC 1A 18 F4 9A 18 :C4
C590: F4 00 00 3C 3F FF 3C 00 00 3E 00 00 0C 30 1C 0C :4C
C5A0: 80 00 8C 00 00 00 3F FF 00 00 00 22 00 00 00 00 :6C
C5B0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C5C0: 22 22 0F FF F0 0C 7F F0 8C 7F F0 03 FF C0 03 9F :1C
C5D0: C0 23 9F C2 00 FF 00 00 FF 00 88 FF 08 00 3C 00 :0D
C5E0: 33 3C CC 00 3C 02 00 FF 00 3F CF FC 80 CF 00 00 :D1
C5F0: 3C 00 33 3C CC 00 3C 02 00 10 00 33 10 CC 80 10 :64
sum : 05 CE 00 46 5F BA 99 79 8A 00 77 00 04 62 F4 81 :20

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: 00 00 8F 00 33 8F CC 00 8F 02 00 FF 00 33 FF CC :DB
C610: 80 FF 00 00 3C 00 3C 00 FC 00 3C 02 00 00 00 00 :70
C620: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C630: 22 22 00 00 00 00 FF FC 88 00 00 30 00 00 30 38 :5F
C640: 0C 32 00 02 3C 00 00 3C FF FC BC 00 00 3F 18 78 :3E
C650: 3F 18 78 3F 18 7A 3F 7E FC 3F 7E FC 8F 7E FC 3F :8A
C660: 7E FC 2F 6E BC 2F 6E BE 3F 18 78 37 18 58 B7 18 :73
C670: 58 3C 00 00 2C FF FC 2E 00 02 30 00 00 30 38 0C :8F
C680: B8 00 00 00 00 00 00 FF FC 22 00 02 00 00 00 00 :D7
C690: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C6A0: 22 22 00 3C 00 3F 3C FC 80 3C 00 00 FF 00 33 FF :E4
C6B0: CC 00 FF 02 00 FF 00 33 CF CC 80 CF 00 00 10 00 :F9
C6C0: 33 10 CC 00 10 02 00 7E 00 33 7E CC 80 7E 00 00 :1A
C6D0: FF 00 3F CF FC 00 CF 02 00 3C 00 33 3C CC 80 3C :0D
C6E0: 00 00 FF 00 00 FF 00 20 FF 22 03 FF C0 03 3F C0 :03
C6F0: 8B 3F C8 0F FF F0 0C FF F0 2C FF F2 00 00 00 00 :A8
sum : 26 14 37 CB B6 66 DA BB 97 3E 1E 25 52 C5 34 1E :6E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C710: 22 22 07 00 E0 00 FF 00 88 FF 08 1C 3C 38 03 00 :4C
C720: C0 23 00 C2 70 3C 0E 0F 00 F0 8F 00 F0 C0 3C 03 :0C
C730: 30 C3 0C 30 C3 0C C0 00 03 3F FF FC 30 3C 0C C0 :33
C740: 00 03 3F FF FC 30 3C 0C C0 3C 03 03 C0 C3 0C C3 :A6
C750: 0C 70 3C 0E 0F 00 F0 0F 00 F0 1C 3C 38 03 00 C0 :17
C760: 83 00 C0 07 00 E0 00 FF 00 20 FF 02 00 00 00 00 :4A
C770: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C780: 22 22 07 00 E0 00 FF 00 88 FF 08 1C 00 38 03 3C :4C
C790: C0 23 00 C2 70 00 0E 0F 3C F0 8F 00 F0 C0 00 03 :A0
C7A0: 30 FF 0C 30 C3 0C CF C3 F3 30 3C 0C 30 3C 0C CF :7E
C7B0: C3 F3 30 3C 0C 30 3C 0C C0 00 03 30 FF 0C 30 C3 :97
C7C0: 0C 70 00 0E 0F 3C F0 0F 00 F0 1C 00 38 03 3C C0 :17
C7D0: 83 00 C0 07 00 E0 00 FF 00 20 FF 02 00 00 00 00 :4A
C7E0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C7F0: 22 22 07 00 E0 00 FF 00 88 FF 08 1C 00 38 03 00 :10
sum : 27 44 58 49 2C B0 98 AD E2 A8 AD FC AE BE F9 3D :02

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: C0 23 3C C2 70 00 0E 0F 00 F0 8F 3C F0 C0 00 03 :0C
C810: 30 C3 0C 30 FF 0C C0 00 03 3F FF FC 30 3C 0C C0 :6F
C820: 00 03 3F FF FC 30 3C 0C C0 00 03 30 C3 0C 30 FF :A6
C830: 0C 70 00 0E 0F 00 F0 0F 3C F0 1C 00 38 03 00 C0 :DB
C840: 83 3C C0 07 00 E0 00 FF 00 20 FF 02 00 00 00 00 :86
C850: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C860: 22 22 07 00 E0 00 FF 00 88 FF 08 1C 00 38 03 3C :4C
C870: C0 23 00 C2 70 00 0E 0F 3C F0 8F 00 F0 C0 00 03 :A0
C880: 30 FF 0C 30 C3 0C CF C0 03 30 3C 0C 30 FF 0C 30 :6F
C890: 00 03 30 3C 0C 3F FF FC C0 00 03 30 FF 0C 30 C3 :A6
C8A0: 0C 70 00 0E 0F 3C F0 0F 00 F0 1C 00 38 03 3C C0 :17
C8B0: 83 00 C0 07 00 E0 00 FF 00 20 FF 02 00 00 00 00 :4A
C8C0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C8D0: 22 22 20 7E 00 60 7E 00 A8 7E 88 23 FF C0 63 FF :B2
C8E0: C0 23 FF C2 2F FF F6 6A FF F2 AA FF F2 3F FF FB :F7
C8F0: 75 FF F9 35 FF F9 3F FF FF 6A FF F9 AA FF F9 3F :1A
sum : 77 90 62 BE D6 DB 79 BB 3C 48 CE DF 1C 0F 02 81 :EB

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: FF FF 75 55 FF 35 55 FF FE 3C 7F EA 28 7D EA 28 :AA
C910: 7D FC 18 36 F5 F7 B2 F4 10 32 23 FF C4 62 AA 84 :11
C920: A2 AA 84 20 00 00 60 00 00 22 22 22 00 00 00 00 :B6
C930: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C940: 22 22 00 7E 00 00 7E 00 88 7E 88 43 FF C0 C3 FF :92
C950: C0 43 FF C2 4F FF F0 CA FF F0 4A FF F0 3F FF FB :2D
C960: 75 FF F9 35 FF F9 3F FF FF 6A FF F9 AA FF F9 1E :F9
C970: 3C 7F 34 14 3F 14 14 3F DC 18 3F F9 EF 9F D8 08 :43
C980: 1F EE 3C 77 FC 14 33 EC 14 33 2B FF CE 3A AA 86 :98
C990: AA AA 86 04 00 00 0C 00 00 26 22 22 00 00 00 00 :54
C9A0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
C9B0: 22 22 00 7E 00 00 7E 00 88 7E 88 03 FF C0 03 FF :92
C9C0: C0 23 FF C2 4F FF F0 CA FF F0 4A FF F0 3F FF FB :0A
C9D0: 75 FF F8 35 FF FA 1C 3C 3B 28 3C 39 88 3C 39 38 :FF
C9E0: 18 1B 33 FF D9 30 18 19 7C 3C 3E 6D EF BE 6C 2C :47
C9F0: 3E 7B FF FC 77 55 DC 73 55 DC 3D FF D8 3F AA C8 :C5
sum : 27 FA 28 1F 1B CA F5 89 27 87 AA 07 80 EE 22 B9 :73

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CA00: BD AA C8 00 80 10 01 80 10 20 A2 32 00 00 00 00 :44
CA10: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CA20: 22 22 00 7E 00 00 7E 00 88 7E 88 43 FF C0 C3 FF :92
CA30: C0 43 FF C2 4F FF F0 CA FF F0 4A FF F0 3F FF FB :2D
CA40: 75 FF F9 35 FF F9 3F FF FF 6A FF F9 AA FF F9 1E :F9
CA50: 3C 7F 34 14 3F 14 14 3F DC 18 3F F9 EF 9F D8 08 :43
CA60: 1F EE 3C 77 FC 14 33 EC 14 33 2B FF CE 3A AA 86 :98
CA70: AA AA 86 04 00 00 0C 00 00 26 22 22 00 00 00 00 :54
CA80: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CA90: 22 22 00 7E 00 00 7E 00 88 7E 88 03 FF C0 03 FF :92
CAA0: CE 73 00 CE 0C 00 30 0F FF F0 8C 00 30 10 00 08 :1D
CAB0: 17 FF F8 10 00 0A 30 00 0C 3B FF FC B0 3C 0C 30 :C2
CAC0: 00 0C 35 FF FC 30 3C 0C 10 00 08 1A FF F8 90 00 :6D
CAD0: 08 0C 00 30 0D 5F 70 0C 00 32 03 00 C0 73 AB CE :8D
CAE0: F3 00 CE 00 56 00 70 7E 0E 72 56 2E 00 00 00 00 :09
CAF0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
sum : 1B D1 B1 8F 74 39 13 BF 3F B6 F9 CE F5 4E F7 11 :B2

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CB00: 22 22 00 7E 00 70 7E 0E 88 7E 00 03 00 C0 73 FF :F9
CB10: CE 03 00 C0 0C 00 30 0F FF F0 8C 00 30 10 00 08 :9F
CB20: 17 FF F8 10 3C 0A 30 00 0C 3B FF FC B0 C3 0C 30 :85
CB30: 00 0C 35 FF FC 30 C3 0C 10 00 08 1A FF F8 90 30 :30
CB40: 08 0C 00 30 0D 5F 70 0C 00 32 03 00 C0 73 AB CE :8D
CB50: 83 00 C0 00 56 00 70 7E 0E 72 56 2E 00 00 00 00 :0D
CB60: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CB70: 22 22 00 7E 00 70 7E 0E 88 7E 00 03 00 C0 73 FF :F9
CB80: CE 03 00 C0 0C 00 30 0F FF F0 8C 00 30 10 00 08 :9F
CB90: 17 FF F8 11 81 9A 30 00 0C 3B FF FC B3 00 CC 30 :5B
CBA0: 00 0C 37 FF FC 33 00 CC 10 00 18 1A FF F8 90 81 :87
CBB0: 98 0C 00 30 0D 7F 70 0C 3C 32 03 00 C0 73 AB CE :79
CBC0: 83 00 C0 00 56 00 70 7E 0E 72 56 2E 00 00 00 00 :0D
CBD0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CBE0: 22 22 00 7E 00 70 7E 0E 88 7E 00 03 00 C0 73 FF :F9
CBF0: CE 03 3C C0 0C 00 30 0F FF F0 8D 81 B0 10 00 18 :ED
sum : A4 9D 18 39 9F 35 FD 53 35 28 75 32 F1 09 A7 32 :8D

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CC00: 17 FF F8 14 00 7A 30 00 0C 3B FF FC B8 00 3C 30 :32
CC10: 00 0C 37 FF FC 34 00 3C 10 00 18 1A FF F8 98 00 :7F
CC20: 78 0C 00 30 0D 7F 70 0D 01 B2 03 00 C0 73 AB CE :9F
CC30: 83 28 C0 00 56 00 70 7E 0E 72 56 2E 00 00 00 00 :35
CC40: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CC50: 22 22 3F FF FC 00 FF 00 80 FF 00 3F E3 FC 03 DF :FC
CC60: C0 03 08 C2 3F 45 FC 0F BA F0 8F 18 F0 3E A2 BC :CC
CC70: 3D 5D 7C 3C 18 3E 20 14 0C 2A EB FC AA C3 FC 10 :72
CC80: 08 0C 15 57 FC 15 43 FE 28 05 7C 2A BA BC A8 18 :DB
CC90: 3C 3A 02 FC 0B 5D F0 0A 18 F2 3E 87 FC 02 B8 C0 :1E
CCA0: 82 9B C0 3F 2F FC 01 2F 00 01 2F 02 00 00 00 00 :A9
CCB0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CCC0: 22 22 3F E7 FC 00 FF 00 80 FF 00 3F E3 FC 03 DF :E4
CCD0: C0 03 08 C2 3F 41 FC 0F BE F0 8F 3C F0 3E 28 BC :76
CCE0: 3D C3 7C 3C C3 3E 00 55 00 3F 00 FC BF 00 FC 00 :04
CCF0: AA 00 3F 00 FC 3F 00 FE 28 14 7C 2A C3 BC A8 C3 :EE
sum : C0 8A 31 B7 E2 DC EA 93 27 42 DE DD FF 1C 52 23 :21

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CD00: 3C 3A 02 FC 0B 7D F0 0A 7C F2 3E 87 FC 02 B8 C0 :A2
CD10: 82 9B C0 3F 27 FC 01 3F 00 01 3F 02 00 00 00 00 :C1
CD20: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CD30: 22 22 3F 00 FC 00 FF 00 80 E7 00 3C 18 3C 03 FF :77
CD40: C0 03 99 C2 30 14 0C 0F C3 F0 8E 00 70 00 AA 00 :08
CD50: 3F 00 FC 38 00 1E 05 55 60 3C 00 3C A4 00 24 06 :91
CD60: AA A0 3C 00 3C 24 00 26 00 55 00 3F 00 FC B8 00 :54
CD70: 1C 30 28 0C 0F C3 F0 0E 00 72 3C 18 3C 03 FF C0 :14
CD80: 83 99 C0 3F 00 FC 00 FF 00 00 E7 02 00 00 00 00 :FF
CD90: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
CDA0: 22 22 00 00 00 3F FF FC BF C3 FC 00 3C 00 3F FF :76
CDB0: FC 30 3C 0E 00 AA 00 3F 00 FC B0 00 0C 01 55 40 :AD
CDC0: 3C 00 3C 30 00 0E 0E AA 80 3C 00 3C 8C 00 30 0D :5F
CDD0: 55 50 3C 00 1C 0C 00 12 02 AA 80 3C 00 3C B0 00 :6F
CDE0: 0C 00 55 00 3F 00 FC 30 00 0E 00 3C 00 3F FF FC :50
CDF0: B0 3C 0C 00 00 00 3F FF FC 3F C3 FC 00 00 00 00 :32
sum : 93 41 CF BE 04 91 49 16 9C 8F 1D 0C 38 B9 B6 11 :91

```



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CE00: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
CE10: 22 22 03 C3 C0 03 C3 C0 88 C8 C8 07 E7 E0 05 E5 :26
CE20: E0 25 E5 E2 03 C3 C0 02 C2 C0 8A C2 C8 00 18 00 :02
CE30: 7F FF FE 00 18 00 00 24 00 1E 3C 78 80 24 00 00 :2E
CE40: 24 00 1E 3C 78 00 24 02 00 18 00 7F FF FE 80 18 :48
CE50: 00 03 C3 C0 03 C3 C0 23 C3 C2 07 E7 E0 05 E5 E0 :4C
CE60: 85 E5 E8 03 C3 C0 02 C2 C0 22 C2 C2 00 00 00 00 :02
CE70: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
CE80: 22 22 03 FF C0 03 C3 C0 88 C3 C8 07 FF E0 05 E5 :72
CE90: E0 25 E5 E2 03 FF C0 02 C2 C0 8A C2 C8 3F FF FC :60
CEA0: 3C 42 3C 42 3C 42 3C 7F FF FE 5E E7 5E E7 5E 3F :75
CEB0: FF FC 2C 42 2C 2C 2C 42 2C 03 FF C0 03 C3 C0 8B C3 :C5
CEC0: C8 07 FF E0 05 E5 E0 25 E5 E2 03 FF C0 02 C2 C0 :AA
CED0: 8A C2 C8 00 00 00 00 00 00 22 22 22 00 00 00 00 :7A
CEE0: 00 00 00 00 00 00 FF FF FF FF FF FF 0F FF 0F FF :FC
CEFO: FF FF 00 00 00 00 00 00 FF FF FF 0F FF 0F FF :07
sum : B8 7B C6 E3 4F 98 9C EE 11 88 74 C3 B5 BF 40 C2 :93

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: F0 5F FF F5 0F FF F0 0A AA B0 AA BA 0E 00 70 :31
CF10: 09 FF 90 58 00 15 0F 00 F0 08 FF 10 A8 C3 1A 0E :A6
CF20: 66 70 09 99 90 59 99 95 00 00 00 03 18 C0 8B 18 :00
CF30: C8 00 00 00 00 00 18 00 22 1A 22 00 00 00 00 00 :3E
CF40: 88 88 88 00 00 00 00 00 00 22 22 22 00 00 00 00 :FE
CF50: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
CF60: 22 22 00 00 00 00 00 00 88 FF 08 00 00 00 00 00 :03
CF70: 00 23 22 E2 00 00 00 00 00 00 8C 88 80 00 00 00 :EB
CF80: 00 00 00 1A 22 1A 00 00 00 00 00 00 98 88 98 00 :0E
CF90: 00 00 00 00 00 1A 22 1A 00 00 00 00 00 00 98 88 :76
CFA0: 98 00 00 00 00 00 00 2E 22 72 00 00 00 00 00 00 :5A
CFB0: 88 89 C8 00 00 00 00 00 00 22 7E 22 00 00 00 00 :9E
CFC0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
CFD0: 22 22 3F C3 FF 00 3C 00 80 3C 00 3F 81 F0 06 7E :71
CFE0: 0F 06 5E 0F 3F C3 C0 03 3C 3F 83 2C 2A 3F E7 80 :41
CFF0: 03 18 7F 03 18 57 00 00 00 FF FF FF FD FF EB 3F :2F
sum : 28 64 26 B7 17 D3 C6 1C 2A 09 5F F3 6A 47 AD 9F :B7

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D000: E7 C0 03 18 3F 03 18 35 3F EF F0 03 7E 0F 83 6E :F0
D010: 0A 3F E7 FC 06 18 03 06 18 03 3F E7 FF 00 18 00 :AB
D020: 80 18 00 00 00 00 00 18 00 22 1A 22 00 00 00 00 :0E
D030: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D040: 22 22 FF C3 FC 00 3C 00 00 3C 00 0F 81 FC F0 7E :7A
D050: 60 F0 5E 60 03 C3 FC FC 3C C0 FC 2C C0 01 E7 FC :94
D060: FE 18 C0 FE 18 C0 00 00 00 FF FF FF FF FF 7F 03 :29
D070: E7 FC FC 18 C0 EC 18 C0 0F EF FC F0 7E C0 B0 6E :C1
D080: C0 3F E7 FC C0 18 60 C0 18 60 FF E7 FC C0 18 00 :4C
D090: 00 18 00 00 00 00 00 18 00 22 1A 22 00 00 00 00 :8E
D0A0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D0B0: 22 22 35 5D 55 30 1C 00 BA BE AA 2E BE AA 0C 3C :77
D0C0: 00 1D 70 55 17 00 FF 03 FF FF AB 00 FF 28 C3 2A :C5
D0D0: 03 FF C0 14 C3 15 00 3C 00 FE FF FF 00 3C 3F 00 :61
D0E0: 1C 00 D5 5F FF 00 1C 3F 14 C3 15 02 EB C0 28 C3 :2E
D0F0: 2A 2B 00 D5 03 55 D5 17 00 D5 1D 69 55 0C 28 00 :52
sum : 03 FD 31 43 0D 3C E7 8C 97 D3 DF D7 34 A5 17 C6 :06

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D100: AE AA AA 3A 96 AA 30 14 00 35 55 55 00 00 00 00 :9F
D110: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D120: 22 22 55 7D 5C 00 3C 0C AA BE AC AA BE 88 00 3C :2A
D130: 30 55 7D 74 FF 00 D4 FF FF C0 FF 00 E8 A8 C3 28 :81
D140: 03 FF 80 54 C3 54 00 3C 00 FE FF FF FC 3C 00 00 :5D
D150: 3C 00 FD 7F FF FC 3C 00 54 C3 54 02 EB 80 A8 C3 :32
D160: 28 55 00 E8 55 55 C0 55 00 D4 55 29 74 00 28 30 :42
D170: AA AA BA 8A 96 AA 00 14 0C 55 55 5C 00 00 00 00 :1E
D180: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D190: 22 22 00 30 C0 00 30 C0 88 B0 C8 0F FF FF 00 30 :61
D1A0: C0 20 30 C0 0F FF FF 00 3F C0 80 3F C0 0F C0 3F :69
D1B0: 00 C0 30 00 FF F0 3F 00 0F 3F 0F 0F BF BF FF 0F :16
D1C0: 00 0C 03 1F CF 03 5F FC 0E 00 0C 02 2F CF 82 AF :A6
D1D0: FC 3F 00 0C 3F 05 0F 3F 55 FC 0F 80 30 00 80 3F :A8
D1E0: 80 AA B0 0F F5 43 00 35 7F 20 35 43 00 00 00 00 :6D
D1F0: 0F E0 05 00 25 55 80 20 05 0F E0 0A 00 2A AA 20 :00
sum : 7E F6 C9 BA 94 8A A8 24 D6 77 84 B1 DE E2 FE 27 :48

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D200: 20 0A 0F FF C1 00 3F F5 80 3F C1 0F C0 30 00 C0 :6C
D210: 3A 00 FF F0 3F 00 0C 3F 0F 00 BF BF FC 0F 00 0C :64
D220: 03 1F CE 03 5F FC 0E 00 0C 02 2F CD 82 AF FC 02 :02
D230: 00 0E 3F 05 0E 3F 55 FE 0F 80 3F 00 80 30 80 AA :9A
D240: 80 0F F5 7F 00 35 40 20 35 40 0F FF FF 00 30 C0 :3A
D250: 80 30 C0 00 30 C0 00 30 C0 22 32 C2 00 00 00 00 :66
D260: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D270: 22 22 03 0C 00 03 0C 00 88 0C 88 FF FF F0 03 0C :7E
D280: 00 03 0C 02 FF FF F0 03 FC 00 03 FC 08 FC 03 F0 :F4
D290: 0C 03 00 0F FF 02 F0 00 FC F0 F0 FC FB FF FC 30 :00
D2A0: 00 F0 F1 FC C0 35 FF C2 20 00 F0 E0 FC C0 2A FF :68
D2B0: C0 30 00 FC F0 50 FC 35 5F FE 08 03 F0 C8 03 00 :B0
D2C0: 0A AB 00 C1 57 F0 FD 54 00 C1 54 02 00 00 00 00 :25
D2D0: F0 0F F0 FF FC 00 F0 0C 00 F0 0F F0 FF FC 00 F0 :C0
D2E0: 0C 02 43 FF F0 7F FC 00 43 FC 00 0C 03 F0 AC 03 :A8
D2F0: 00 0F FF 02 30 00 FC 70 F0 FC 3B FF FC 30 00 F0 :EE
sum : 81 89 02 4C BE 28 42 D4 5C D3 40 33 A9 DD 87 A5 :A8

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D300: B1 FC C0 35 FF C2 20 00 F0 62 FC C0 2A FF C0 B0 :2A
D310: 00 FC B0 50 FC B5 5F FC F8 03 F0 08 03 00 0A AB :B3
D320: 00 FF 57 F0 03 54 00 03 54 02 FF FF F0 03 0C 00 :F3
D330: 03 0C 00 03 0C 00 03 0C 00 23 2C 22 00 00 00 00 :9E
D340: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D350: 22 22 FF FF FF 0F FF FF 0F FF FF F0 C3 0C 30 C3 :0D
D360: 0C 30 C3 0C EE 00 07 EF FF FF EE 00 07 D8 00 05 :BF
D370: DA 3F FD D8 C0 05 A0 00 05 B5 0F FD A0 30 05 00 :EE
D380: 00 00 18 A3 FF 03 0C 07 A5 80 07 B5 D0 F8 A5 83 :A4
D390: 18 E5 80 3C F5 AA 03 E5 80 C3 A5 80 C0 B5 07 0F :03
D3A0: A5 83 1F 05 81 C0 15 AE 37 05 86 37 00 00 00 00 :49
D3B0: A5 83 80 B5 AC 6F A5 8C 6F E5 83 80 F5 DC 57 E5 :0D
D3C0: 8C 57 A5 81 C0 B5 AE 2A A5 86 2A 05 80 C0 15 D7 :0C
D3D0: 05 05 83 35 A5 80 1C B5 AA 02 A5 80 C2 E5 80 07 :B7
D3E0: F5 D4 58 E5 83 18 A0 00 00 BB A0 AF A3 0C 07 00 :01
D3F0: 00 05 15 45 55 00 30 05 B8 00 05 BA 0A AD B8 C0 :8F
sum : A4 B4 55 D4 15 08 13 88 A9 AD 3C B0 FB FD 32 5A :02

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D400: 05 EE 00 07 EF 55 57 EE 00 07 F0 C3 0C 30 C3 0C :48
D410: 30 C3 0C FF FF FF FF FF FF 0F FF FF 00 00 00 00 :16
D420: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D430: 22 22 FF FF FF FF FF F0 FF FF F0 30 C3 0F 30 C3 :12
D440: 0C 30 C3 0C E0 00 77 FF FF F7 E0 00 77 E0 00 18 :A9
D450: FF FC 7B E0 03 18 E0 00 05 FF F1 FD E0 0C 05 00 :37
D460: 00 00 FF C6 F8 E0 30 C0 E0 01 E5 1F 1D FD 18 C1 :65
D470: E5 3C 01 67 C0 7D 7F C3 01 67 03 01 65 F0 FD 7D :43
D480: F8 C1 65 03 81 60 FC 7D 78 FC 61 60 00 00 00 00 :80
D490: 01 C1 65 FE 3D 7D FE 31 65 01 C1 67 FE 3D 7F FE :54
D4A0: 31 67 03 81 65 FC 7D 7D FC 61 65 03 01 60 50 FD :EA
D4B0: 78 54 C1 60 38 01 65 80 FD 7D 83 01 65 E0 01 67 :86
D4C0: 1C 3D 7F 18 C1 67 00 00 05 EA 8E FD E0 30 C5 E0 :47
D4D0: 00 00 F5 43 F8 E0 0C 00 E0 00 1D EA A8 FD E0 03 :88
D4E0: 1D E0 00 77 F5 55 F7 E0 00 77 30 C3 0F 30 C3 0C :0D
D4F0: 30 C3 0C FF FF FF FF FF F0 FF FF F0 00 00 00 00 :08
sum : 52 58 57 D1 90 40 D1 71 16 AE 7C 74 A3 F2 45 9B :0D

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D500: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D510: 22 22 00 00 00 00 31 59 20 88 98 88 00 00 00 8F :E4
D520: F6 20 3C 00 00 66 00 37 99 F0 88 18 08 01 C3 80 :64
D530: 4A 3C 4C 22 3C 42 00 00 00 1F FF F8 9F FF F8 01 :1F
D540: C3 80 12 3C 58 22 3C 42 00 66 00 C7 19 E2 08 18 :D1
D550: 08 00 00 00 1F BF DC 20 3C 22 00 00 00 63 B9 E3 :3F
D560: 88 18 08 00 00 00 0C 81 30 22 22 02 00 00 00 00 :AB
D570: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D580: 22 22 00 7E 00 20 7E 02 9E 7E 7C 01 EF 80 91 EF :EA
D590: 81 21 EF 8C 02 EF C0 36 EF EC 82 EF C0 03 5F C0 :32
D5A0: 5F 5F FA 17 5F 00 00 00 00 FE AF FF 2A AA AB 00 :26
D5B0: 00 00 BF 5F 7D 15 55 50 02 AB C0 6E AB FE 0A AB :8E
D5C0: C8 03 6D 40 9B 6D 6C 03 6D 40 01 AA 80 25 AA 85 :1B
D5D0: 91 AA 88 00 56 00 08 56 00 36 56 7C 00 00 00 00 :7F
D5E0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D5F0: 22 22 00 00 00 02 7E 20 88 00 08 01 E7 80 15 FF :F0
sum : 32 87 3F 1E 82 1D 9A D3 A9 DA FD 5D AB 15 6C 7F :AA

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D600: 88 01 E7 82 82 A7 C1 8F FF F1 82 A7 C1 61 45 46 :31
D610: 7F FF FE 61 45 46 F8 00 1F FF FF FF F8 00 1F 61 :F4
D620: 45 C6 7F FF FE 61 45 C6 82 A7 C1 8F FF F1 82 A7 :85
D630: C1 01 65 80 2D FF 96 01 65 80 C0 00 00 08 7E 08 :D0
D640: 80 00 00 00 00 00 00 81 00 22 22 22 00 00 00 00 :67
D650: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D660: 22 22 00 00 00 01 FF 80 89 FF 88 08 00 10 35 7F :A0
D670: 6C 35 7F 6C 30 00 08 CA BF BF CA BF B7 20 00 04 :68
D680: 94 FF 39 D4 18 38 20 00 04 99 EF 99 D8 2C 18 20 :77
D690: 00 04 95 55 F9 D5 55 FB 10 00 0C 4F FF F2 6C 00 :D4
D6A0: 32 0C 00 30 33 FF CC 33 81 CC 01 C3 80 40 3C 02 :AE
D6B0: 7E 00 7E 00 00 00 55 42 AA 7F C3 FE 00 00 00 00 :7D
D6C0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D6D0: 22 22 01 81 80 01 99 80 9F FF F8 01 C7 80 01 99 :D8
D6E0: 80 19 99 9A 68 A7 FE 68 99 FE 68 99 FE 05 47 E0 :0C
D6F0: 00 18 00 10 18 1A 00 00 00 1A BF F8 9A BF F8 01 :7D
sum : 01 80 2E 52 69 1F D8 8C D4 EA 97 59 25 2C 9C B9 :41

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D700: 45 E0 00 18 00 14 18 1A 6A A3 FE 6A 89 FE 6A 89 :72
D710: FE 01 45 80 01 19 80 15 19 9A 00 80 80 00 98 80 :3E
D720: 9A 9A F8 01 81 80 01 81 80 21 A1 A2 00 00 00 00 :9A
D730: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D740: 22 22 1F FF F8 1F FF F8 1F FF F8 50 1C 0F 50 1C :6D
D750: 0F 55 5F FF A0 2C 0F A0 2C 0F AA AF FF 1F 0F F8 :C6
D760: 10 1C 08 10 1C 08 A0 2C 0F A0 2C 0F AA AF FF 50 :C6
D770: 1C 0F 50 1C 0F 55 5F FF 1F EF F8 10 2C 08 10 2C :0F
D780: 08 50 1C 0F 50 1C 0F 55 5F FF A0 2C 0F A0 2C 0F :67
D790: AA AF FF 15 50 58 15 50 58 15 50 58 00 00 00 00 :86
D7A0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :8A
D7B0: 22 22 00 00 00 00 FF 00 88 FF 08 00 00 00 07 FF :D8
D7C0: E0 23 FF C2 00 00 00 1B FF F8 98 FF 38 00 00 00 :A5
D7D0: 35 FF FC 34 7E 7E 00 00 00 2B FF FC AA 7E 7C 00 :2A
D7E0: 00 00 15 FF FC 14 7E 7E 00 00 00 2B BF FC AA 3E :EE
D7F0: 7C 00 00 00 17 55 F8 14 55 3A 00 00 00 06 AA E0 :13
sum : 9F 60 3E DC 83 80 4F E2 1F 68 01 54 AA 03 43 09 :55

```



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D800: 82 AA C8 00 00 00 00 05 00 20 05 22 00 00 00 00 :E0
D810: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
D820: 22 22 00 FF 00 00 FF 00 88 FF 08 07 FF E0 07 00 :BE
D830: E0 27 00 E2 18 FF F8 18 7E 18 98 7E 18 35 01 FC :09
D840: 31 01 8C 31 7F 8C 2A 3C FC 22 3C CC A2 BF CC 36 :E9
D850: 1C FC 32 1C CC 32 DF CC 2B 00 BC 21 00 8C A1 2A :6E
D860: 8C 15 55 78 10 54 18 10 54 1A 06 AA E0 06 00 E0 :DE
D870: 86 00 E8 00 05 00 00 05 00 22 05 22 00 00 00 00 :31
D880: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
D890: 22 22 3F 00 00 3F 00 00 BF 08 88 7E 3C 00 55 BD :DD
D8A0: 80 54 3C 22 7F 99 FC 3D FF BC BD 99 BC 7F E7 FE :B4
D8B0: 7B 7F DE 7B 67 DE 47 FF E2 7A BF DE C2 BF C2 FF :19
D8C0: 99 FE FD FF BE FD 99 BE E7 FF FE E2 7E 7C E2 7E :C5
D8D0: 7C 78 18 FE 78 18 AA 7A 3A AA 00 00 FE 00 00 D6 :76
D8E0: 88 88 D6 00 00 FC 00 00 FC 22 22 FC 00 00 00 00 :1E
D8F0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
sum : 9D F8 07 40 67 D8 36 E6 D6 9E 6C 33 CF 20 55 B0 :3E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D900: 22 22 00 00 FC 00 00 FC 88 88 FC 00 3C 7E 01 BD :C0
D910: AA 20 3C 2A 3F 99 FE 3D FF BC BD 99 BC 7F E7 FE :74
D920: 7B 7F DE 7B 67 DE 47 FF E2 7A BF DE C2 BF C2 7F :99
D930: 99 FF 7D FF BF 7D 99 BF 7F FF E7 3E 7E 47 BE 7E :4C
D940: 47 7F 18 1E 55 18 1E 55 3A 3E 7F 00 00 68 00 00 :3E
D950: E8 88 88 3F 00 00 3F 00 00 3F 22 22 00 00 00 00 :FC
D960: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
D970: 22 22 00 00 00 1F FF F8 9F FF F8 0F E7 F0 30 18 :1E
D980: 0C 30 18 0E 18 00 18 60 18 06 67 FF E6 3C 00 3C :D4
D990: D8 18 18 D8 FF DB FE 00 7F FD FF BF FD FF BF FE :B1
D9A0: 00 7F FD 55 BF FD 55 BF 3C 00 3C 98 08 18 9A AA :18
D9B0: D8 10 00 18 40 10 06 45 55 46 0A A2 A0 20 08 0C :B9
D9C0: A0 08 0C 00 00 00 15 55 58 35 55 5A 00 00 00 00 :5A
D9D0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
D9E0: 22 22 00 00 00 1F FF F8 9F FF F8 00 00 00 3F FF :2E
D9F0: FC 30 18 0E 00 00 00 78 18 1E 67 FF E6 18 00 18 :7C
sum : B1 1A 88 65 CC 32 CF 7D 08 D4 58 37 90 EC 38 1B :3F

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DA00: FC 18 3F DB FF DB FC 00 3F FF FF FF FD FF BF FC :F7
DA10: 00 3F FF 55 FF FD 55 BF 18 00 18 BC 08 3F 9A AA :1A
DA20: DB 00 00 00 50 10 1E 45 55 46 00 00 00 2A AA AC :B9
DA30: A0 08 0C 00 00 00 15 55 58 35 55 5A 00 00 00 00 :5A
DA40: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DA50: 22 22 00 03 C0 00 03 C0 88 88 C8 00 00 78 00 00 :1D
DA60: 78 22 22 7A 00 00 1E 00 00 1E 88 88 9E 00 00 0F :2F
DA70: 00 00 0F 22 22 2F 00 00 07 00 00 07 88 88 87 00 :27
DA80: 00 07 00 00 07 22 22 27 00 00 0F 00 00 0F 88 88 :A7
DA90: 8F 00 00 1C 00 00 1C 22 22 1E 00 00 78 00 00 78 :19
DAA0: 88 88 78 00 03 C0 00 03 C0 22 23 C2 00 00 00 00 :15
DAB0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DAC0: 22 22 00 00 00 00 00 00 88 88 88 00 3F 00 00 3F :5A
DAD0: 00 22 3F 22 01 E3 E0 01 E3 E0 89 EB E8 0F 00 7C :F2
DAE0: 0F 00 7C 2F 22 7C 38 00 1F 38 00 1F 88 88 9F 38 :1D
DAF0: 00 1F 38 00 1F 3A 22 1F 0F 00 7C 0F 00 7C 8F 08 :9E
sum : 59 95 E6 3C 7C 92 2D 95 1E 03 7B 7F 82 8A 40 A0 :E7

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DB00: 7C 01 E3 E0 01 E3 E0 21 E3 E2 00 3F 00 00 3F 00 :68
DB10: 88 BF 08 00 00 00 00 00 00 22 22 22 00 00 00 00 :B5
DB20: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DB30: 22 22 03 C0 00 03 C0 00 88 C8 88 1E 00 00 1E 00 :E1
DB40: 00 1E 22 22 78 00 00 78 00 00 78 88 88 F0 00 00 :CA
DB50: F0 00 00 F2 22 22 E0 00 00 E0 00 00 E8 88 88 E0 :BE
DB60: 00 00 E0 00 00 E2 22 22 F0 00 00 F0 00 00 F0 88 :5E
DB70: 88 38 00 00 38 00 00 3A 22 22 1E 00 00 1E 00 00 :B2
DB80: 9E 88 88 03 C0 00 03 C0 00 23 C2 22 00 00 00 00 :38
DB90: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DBA0: 22 22 00 00 00 00 00 00 88 88 88 00 FC 00 00 FC :D4
DBB0: 00 22 FC 22 07 C7 80 07 C7 80 87 C7 88 3E 00 F0 :E0
DBC0: 3E 00 F0 3E 22 F2 F8 00 1C F8 00 1C F8 88 9C F8 :BC
DBD0: 00 1C F8 00 1C FA 22 1E 3E 00 F0 3E 00 F0 BE 88 :0C
DBE0: F0 07 C7 80 07 C7 80 27 C7 A2 00 FC 00 00 FC 00 :14
DBF0: 88 FC 88 00 00 00 00 00 00 22 22 22 00 00 00 00 :72
sum : 14 23 AB 97 DF 64 CF 11 00 B5 23 58 EC 4C 2B 18 :47

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DC00: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DC10: 22 22 00 00 00 00 18 00 88 98 88 00 00 00 00 3C :40
DC20: 00 22 3C 22 00 00 00 00 5E 00 88 5E 08 00 00 00 :CC
DC30: 02 BF C0 22 BF C2 00 00 00 1D 5F F8 9D 5F F8 00 :8C
DC40: 00 00 02 AB C0 22 AB C2 00 00 00 00 00 00 56 88 :30
DC50: 08 00 00 00 00 00 2C 00 22 2C 22 00 00 00 18 00 :8C
DC60: 88 98 88 00 00 00 00 00 00 22 22 22 00 00 00 00 :0E
DC70: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DC80: 22 22 00 7E 00 00 7E 00 BE 7E 7C 01 EF 80 01 EF :58
DC90: 80 31 EF 8C 02 EF C0 02 EF C0 B2 EF CC 03 5F C0 :1D
DCA0: 03 5F C0 33 5F CC 02 AF C0 00 00 00 80 00 0C 03 :80
DCB0: 5F 40 00 00 00 30 00 0C 02 AB C0 02 AB C0 B2 AB :12
DCC0: CC 03 6D 40 03 6D 40 33 6D 4C 01 AA 80 01 AA 80 :6E
DCD0: B1 AA 8C 00 56 00 00 56 00 3E 56 7C 00 00 00 00 :A3
DCE0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DCF0: 22 22 03 FF C0 00 00 00 88 00 08 0F FF F0 0C 3C :DC
sum : 57 5C 31 6B F9 68 DB C2 0E 6C DE 9F 90 93 6C 11 :E4

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DD00: 30 2C 3C 32 3F FF FC 30 FF 0C BQ FF 0C 3F 7F FC :B4
DD10: 33 FF CC 33 7F CC AA FF FF FF FF FF AA FF FF 3D :06
DD20: 5D FC 33 FF CC 31 5D CC 3F AB FC 30 FF 0C B0 AB :2D
DD30: 0C 0F 07 F0 0C 3C 30 2C 14 32 03 FF C0 00 00 00 :8E
DD40: 88 00 08 00 00 00 00 00 00 22 22 22 00 00 00 :F6
DD50: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DD60: 22 22 00 00 00 01 FF 80 89 FF 88 0C 00 30 33 7F :C2
DD70: 4C 33 7F 4C 18 00 18 E6 8F A7 E6 BF A7 18 00 18 :42
DD80: A5 7F 65 E5 7F 67 18 00 18 A6 DB 65 E6 18 67 0C :DB
DD90: 00 30 93 55 C9 F3 55 CF 07 00 E0 48 AB 12 78 AB :07
DDA0: 1E 03 FF C0 34 00 2C 3C 00 3C 00 AA 80 41 55 42 :BA
ddb0: 7F 55 7E 00 00 00 55 42 AA 7F C3 FE 00 00 00 00 :D3
DDC0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DDD0: 22 22 00 03 FF 00 03 FF 88 88 FF FF FF FF FF FF :55
DDE0: 55 FF FF 55 7F FF FF 60 03 55 60 03 55 3F FF FF :D2
DDF0: 30 01 FF 30 01 FF 1F FF FF 1F FF FF 9F FF FF 15 :4C
sum : AB B4 0C 22 A9 91 69 48 FC 10 1A 70 20 3A 92 CB :C5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DE00: 55 FF 15 55 FF 15 55 FF 3A AB 55 30 01 55 80 01 :97
DE10: 55 55 57 55 40 03 55 40 03 55 AA AB 55 AA AB 55 :DA
DE20: AA AB 55 00 03 55 00 03 55 22 23 55 00 00 00 00 :F4
DE30: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DE40: 22 22 FF C0 00 FF C0 00 FF C8 88 FF FF FF AA FF :B7
DE50: FF AA FF FF FF FF FE AA C0 06 AA C0 06 FF FF FC :7D
DE60: FF 80 0C FF 80 0C FF FF F8 FF FF F8 FF FF F8 :F7
DE70: D5 58 FF D5 58 FF D5 5A AA AA AC AA 80 0C AA 80 :E7
DE80: 0C AA D5 56 AA C0 06 AA C0 06 AA EA AB AA EA AB :3F
DE90: AA EA AB AA C0 00 AA C0 00 AA C2 22 00 00 00 00 :A1
DEA0: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DEB0: 22 22 00 00 00 FF FC 00 FF FC 88 00 00 00 FF FC :BD
DEC0: 00 FF FE 22 30 00 00 BF FF FF 30 00 00 0C 00 00 :48
DED0: 5F FF FF 0C 00 00 03 FF FF AB FF FF 03 FF FF 0C :20
DEE0: 00 00 55 55 55 0C 00 00 30 00 00 BA AA AA 30 00 :81
DEF0: 00 00 00 00 D5 54 00 D5 56 22 00 00 00 AA AC 00 :CC
sum : 80 57 A4 C0 DD 95 FB 52 46 11 22 56 32 11 6A C7 :3D

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DF00: AA AC 88 00 00 00 00 00 00 22 22 22 00 00 00 :44
DF10: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
DF20: 22 22 00 00 00 00 3F FF 88 BF FF 00 00 00 00 3F :07
DF30: FF 22 3F FF 00 00 0C FF FF FF 00 00 0C 00 00 30 :A4
DF40: FF FF FF 00 00 30 FF FF C0 FF FF FF FF FF C0 00 :A6
DF50: 00 30 55 55 7F 00 00 30 00 00 0C AA AA AF 00 00 :98
DF60: 0C 00 00 00 00 15 57 22 15 57 00 00 00 00 2A AB :DB
DF70: 88 AA AB 00 00 00 00 00 00 22 22 22 00 00 00 :43
DF80: 00 00 00 00 00 88 88 88 00 00 00 00 00 00 22 :BA
DF90: 22 22 00 55 5F 00 00 0F 88 AA AF 00 AA 85 00 00 :47
DFA0: 15 22 55 55 00 D5 6A 00 00 2A 88 2A AA 3F EA D5 :A4
DFB0: 00 00 55 00 15 55 2F F5 FF 00 00 FF 80 0A FF 15 :7F
DFC0: FB FF 00 01 FF 00 05 FF 2A BF FF 00 03 00 80 03 :6C
DFD0: 00 15 57 80 00 06 7F 00 06 7F 2A AF 00 00 0C FF :DA
DFE0: 80 0C 80 15 5E 00 00 19 EA 00 19 00 00 00 00 00 :98
DFF0: 2A BE 00 00 19 D5 80 19 00 15 5E 00 00 19 D5 00 :D0
sum : 3A EB 47 94 69 4A 4E 94 0D 7F 25 C5 8C C5 34 4A :DA

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
E000: 19 00 2A BE 00 00 19 EA 80 19 00 15 5F 00 00 0C :1D
E010: FF 00 0C 80 2A AF 80 00 06 7F 80 06 7F 15 5F FF :E1
E020: 00 03 00 00 03 00 2A FB FF 00 01 FF 80 05 FF 17 :C5
E030: F5 FF 00 00 FF 00 0A FB 3F EA AA 00 00 2A 80 15 :8E
E040: 2A 00 D5 55 00 00 15 22 2A 95 00 AA AA 00 00 0A :A8
E050: 88 55 4A 00 55 5F 00 00 0F 22 AA AF 00 00 00 00 :65
E060: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
E070: 22 22 FA AA 00 F0 00 00 F5 55 88 5D 55 00 58 00 :B4
E080: 00 5A AA 22 AE AB 00 AC 00 00 AD 54 88 57 57 FC :5E
E090: 56 00 00 56 AB 00 FF AF E8 FF 00 00 FF 50 00 FF :37
EOA0: DF 54 FF 80 00 FF A0 00 FF FA AB 00 C0 00 00 C0 :72
EOB0: 00 01 F5 54 FE 60 00 FE 60 00 00 FA AB FF 30 00 :D7
EOC0: 01 30 00 00 7D 54 57 98 00 00 98 00 00 00 00 :89
EOD0: 00 7D 54 AB 98 00 00 98 00 00 7A AB AB 98 00 00 :11
EOE0: 98 00 00 7D 54 57 98 00 00 98 00 00 FA AB FF 30 :C1
EOF0: 00 01 30 00 01 F5 54 FE 60 00 FE 60 00 FF FA AB :D8
sum : AF D6 71 B1 3F AB 4C 15 21 1F C2 26 F1 29 B6 F6 :D0

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
E100: 00 C0 00 00 C0 00 FF DF 54 FF 80 00 FF A0 00 FF :CF
E110: AF E8 FF 00 00 FF 50 00 AB 57 FC AA 00 00 AA AB :DF
E120: 00 56 AB 00 54 00 00 55 54 22 AD 55 00 AB 00 00 :CA
E130: AA AA 88 FA AA 00 F0 00 00 F5 55 22 00 00 00 :DC
E140: 00 00 00 00 00 00 88 88 88 00 00 00 00 00 22 :BA
E150: 22 22 00 00 00 00 01 80 9F E1 87 07 FE 7F 00 01 :51
E160: 80 18 01 80 02 00 00 01 FF FF 99 FF FF 04 7F FE :32
E170: 01 00 01 19 00 01 02 20 1E 00 80 01 80 9F E1 04 :E1
E180: 47 FF 01 00 00 01 10 00 00 23 FF 1A 81 FF 9A 89 :37
E190: FF 04 45 7F 01 01 7F 01 11 7F 02 2A BF 00 80 BF :03
E1A0: 80 80 BF 04 55 57 01 01 57 19 01 57 00 00 00 :39
E1B0: 02 02 AB 00 AB AB 98 AB AB 04 55 57 01 01 57 19 :0F
E1C0: 01 57 02 2A BA 00 80 BA 80 80 BA 04 45 75 01 01 :F2
E1D0: 75 01 11 75 00 23 AA 1A 81 AA 9A 89 AA 04 47 55 :7B
E1E0: 01 00 00 01 10 00 02 20 0A 00 80 01 80 9A A1 04 :7E
E1F0: 75 54 01 00 01 19 00 01 02 00 00 01 AA AA 99 AA :7F
sum : 80 13 F8 B6 89 40 1E FC R7 36 49 A9 D6 27 FD 31 :5E

```



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
E200:	AA	07	54	55	00	01	00	18	01	00	00	00	00	00	01	80	:F5
E210:	9F	E1	87	00	00	00	00	00	00	22	22	22	00	00	00	00	:6D
E220:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:8A
E230:	22	22	00	00	00	01	80	00	E1	87	F8	FE	7F	C0	01	80	:E3
E240:	00	01	80	1A	00	00	60	FF	FF	00	FF	FF	18	7F	FC	60	:EA
E250:	80	01	80	80	01	9A	78	06	60	80	01	80	87	F1	80	FF	:F2
E260:	C6	60	00	01	80	00	19	82	FF	66	00	FF	01	F8	FF	19	:B7
E270:	F8	FD	E6	60	FD	81	80	FD	99	82	F7	FE	60	F7	81	80	:9E
E280:	F7	81	80	DF	FE	60	DF	81	80	DF	81	9A	00	00	00	00	:0F
E290:	7F	E0	60	7F	9F	80	7F	9F	98	7F	FE	60	7F	81	80	7F	:EF
E2A0:	81	9A	AF	FE	60	AF	81	80	AF	81	80	57	E6	60	57	81	:FD
E2B0:	80	57	99	82	AA	E6	00	AA	81	F8	AA	99	F8	55	A6	60	:3B
E2C0:	00	01	80	00	19	82	28	06	60	80	01	80	82	A9	80	55	:AB
E2D0:	52	60	80	01	80	80	01	9A	00	00	60	AA	AA	80	AA	AA	:56
E2E0:	98	54	55	20	01	00	00	01	00	1A	00	00	00	01	80	00	:FE
E2F0:	E1	87	F8	00	00	00	00	00	00	22	22	22	00	00	00	00	:C6
sum :	EB	F7	36	4F	BF	94	81	0F	09	A4	3D	D2	08	7F	25	79	:2B



## リスト7-6 アイテムキャラクタ "DATA3"

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C000:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C010:	22	22	00	00	00	3F	FF	FC	BF	FF	FC	1F	FF	F8	20	00	:6E
C020:	04	20	00	06	1F	FF	F8	27	E7	C4	A7	E7	C4	1F	FF	F8	:7A
C030:	21	86	64	21	86	66	1F	FF	F8	21	87	C4	A1	87	C4	1F	:A5
C040:	FF	F8	21	86	64	21	86	66	1F	FF	F8	21	87	C4	A1	87	:B9
C050:	C4	1F	FF	F8	20	00	04	20	00	06	00	00	00	3F	FF	FC	:5E
C060:	BF	FF	FC	00	00	00	00	00	00	22	22	22	00	00	00	00	:20
C070:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C080:	22	22	00	00	00	3F	FF	FC	BF	FF	FC	0A	AA	A8	35	55	:1E
C090:	54	20	00	06	17	D7	D0	28	EF	EC	A3	C7	C4	0E	EE	E8	:50
C0A0:	37	77	74	26	66	66	17	57	D0	2E	AF	EC	A6	07	C4	0E	:9A
C0B0:	EE	E8	37	77	74	26	66	66	17	D7	D0	2B	EF	EC	A3	C7	:18
C0C0:	C4	0A	AA	A8	35	55	54	20	00	06	00	00	00	3F	FF	FC	:5E
C0D0:	BF	FF	FC	00	00	00	00	00	00	22	22	22	00	00	00	00	:20
C0E0:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C0F0:	22	22	00	00	00	3F	FF	FC	BF	FF	FC	00	00	00	3F	FF	:76
sum :	09	AA	D1	F0	4F	FB	D7	40	A9	22	80	17	EE	89	4B	0D	:06

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C100:	FC	20	00	06	03	E7	C0	3F	FF	FC	A3	E7	C4	06	06	60	:C0
C110:	3F	FF	FC	26	06	66	03	C7	C0	3F	FF	FC	A3	C7	C4	00	:BE
C120:	66	60	3F	FF	FC	20	66	66	07	C7	C0	3F	FF	FC	A7	C7	:22
C130:	C4	00	00	00	3F	FF	FC	20	00	06	00	00	00	3F	FF	FC	:5E
C140:	BF	FF	FC	00	00	00	00	00	00	22	22	22	00	00	00	00	:20
C150:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C160:	22	22	00	00	1F	00	00	1F	88	88	9F	00	00	0E	1F	FF	:50
C170:	FE	00	00	0E	C7	F9	BC	FF	FF	FC	C7	F9	BC	00	00	0E	:0C
C180:	1F	FF	FE	00	00	0E	00	00	1F	00	00	1F	88	88	9F	F8	:0F
C190:	00	00	F8	00	00	FA	22	22	70	00	00	7F	FF	F8	70	00	:8C
C1A0:	00	3C	FF	F3	3F	FF	FF	3C	FF	F3	70	00	00	7F	FF	F8	:7F
C1B0:	70	00	00	F8	00	00	F8	00	00	FA	22	22	00	00	00	00	:9E
C1C0:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C1D0:	22	22	00	00	00	00	18	00	88	98	88	00	00	00	00	7E	:82
C1E0:	00	22	7E	22	00	00	00	00	BF	00	88	BF	08	00	00	00	:00
C1F0:	01	5F	80	21	5F	A2	00	B5	00	01	0A	80	89	0A	88	00	:5D
sum :	F6	7E	2A	67	C8	0E	22	CD	32	34	96	3C	3A	1F	25	E2	:62

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C200:	00	00	01	5F	80	21	5F	A2	00	85	00	01	0A	80	89	0A	:05
C210:	88	00	00	00	03	5F	C0	23	5F	C2	00	00	00	06	AF	E0	:83
C220:	86	AF	E8	00	00	00	00	81	F0	2D	81	F2	00	00	00	00	:3B
C230:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C240:	22	22	3F	FF	FC	3F	FF	FC	BF	FF	FC	6A	AA	AA	40	00	:70
C250:	02	55	55	56	D0	14	15	87	C3	C1	AF	EB	EB	A8	88	88	:46
C260:	86	66	61	D7	77	75	00	11	55	87	C6	01	AF	EE	AB	A8	:84
C270:	88	88	86	66	61	D7	77	75	D0	14	15	87	C3	C1	AF	EB	:C1
C280:	EB	6A	AA	AA	40	00	02	55	55	56	3F	FF	FC	3F	FF	FC	:5F
C290:	BF	FF	FC	00	00	00	00	00	00	22	22	22	00	00	00	00	:20
C2A0:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C2B0:	22	22	00	FF	00	00	FF	00	88	FF	88	0E	00	70	0F	FF	:DD
C2C0:	F0	2E	00	72	08	81	10	0F	FF	F0	88	81	18	08	42	10	:A2
C2D0:	0F	FF	F0	28	42	12	08	24	10	0A	FF	F0	88	24	18	08	:7B
C2E0:	18	10	0A	FF	F0	28	18	12	04	66	20	05	7F	E0	84	66	:4B
C2F0:	28	02	00	40	02	AF	C0	22	00	62	00	81	00	00	D5	00	:85
sum :	4B	E1	04	73	A3	89	88	1B	F6	38	97	F6	2C	42	1B	C5	:7B

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:sum
C300:	88	81	88	00	2C	00	00	2C	00	22	2E	22	00	00	00	00	:5B
C310:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C320:	22	22	00	00	00	00	3C	00	88	BC	88	00	18	00	01	E7	:4C
C330:	80	21	E7	A2	00	18	00	0F	E7	F0	8F	E7	F8	03	FF	C0	:58
C340:	0C	00	30	2C	00	32	00	18	00	0A	E7	F0	8A	E7	F8	00	:FC
C350:	18	00	0A	E7	F0	2A	E7	F2	00	18	00	05	67	E0	85	67	:4C
C360:	E8	00	18	00	02	A7	C0	22	A7	E2	00	18	00	00	C5	00	:F1
C370:	88	C5	88	00	00	00	00	2C	00	22	2E	22	00	00	00	00	:73
C380:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
C390:	22	22	00	00	00	00	00	00	88	88	88	00	00	00	1F	FF	:FA
C3A0:	F8	00	00	00	00	00	00	7F	FF	FE	00	00	00	1E	97	A0	:C9
C3B0:	FF	FF	FF	1E	97	A0	10	94	20	FF	FF	FF	10	94	20	1C	:F3
C3C0:	97	20	FF	FF	FF	1C	97	20	10	94	20	FF	FF	FF	10	94	:EC
C3D0:	20	10	67	BC	FF	FF	FF	10	67	BC	00	00	00	7F	FF	FE	:FF
C3E0:	00	00	00	00	00	00	1F	FF	F8	00	00	00	00	00	00	00	:16
C3F0:	00	00	00	00	00	00	88	88	88	00	00	00	00	00	00	22	:BA
sum :	8E	DA	AE	8E	B3	D6	40	6D	C4	C9	01	36	10	FA	27	C1	:90





## リスト7-7 戦士キャラクタデータ "DATA 4"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: 00 00 00 00 00 60 00 00 00 00 00 60 3F 80 90 3F :4E
C010: 80 F0 7F C0 98 6F 40 F8 7F FF 98 57 18 78 7F FF :69
C020: FC 3E 7E 7C 7F 99 FE 7D FF BE 47 FF E2 7A BF DE :C3
C030: 7F E7 FE 7B 7F DE 3F 99 FE 29 FF 94 00 7E 7E 01 :CB
C040: AB AE 00 01 FC 00 01 5C 00 00 00 00 00 00 00 :B3
C050: 00 00 00 00 00 60 00 00 00 00 00 60 01 9C 00 :5D
C060: 01 FC 1E 03 9E 1E 02 FE 3F FF 9E 3C 19 76 3F FF :BF
C070: FE 2A 7E 7C 7F 99 FE 7D FF BE 47 FF E2 7A BF DE :B1
C080: 7F E7 FE 53 7F DE 7F 99 F8 3D FF A8 7E 7E 00 55 :59
C090: AB 80 3F 80 00 3A 80 00 00 00 00 00 00 00 00 :A4
C0A0: 00 00 00 00 00 00 00 00 00 00 00 33 EE 03 FE 7F 03 :E1
C0B0: CF F5 00 1F FF FF FB EB 03 FF F3 03 2F FD 00 33 :1E
C0C0: CE 00 1F FE 00 3F 3C 00 2F FC 07 F3 CC 07 9F FC :F9
C0D0: 0F FF F0 0D 6D 7C 0F FF F8 0F BA E8 0F FE 7D 0D :35
C0E0: 57 F0 07 FF E0 06 B3 E0 00 00 00 00 00 00 00 :C6
C0F0: 00 00 00 00 00 00 07 FF E0 07 DB E0 0F FE 7D 0B :30
sum : D2 34 EA 33 7A 35 7D 86 BA F1 84 39 CE 5D E4 99 :E5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C100: F7 F0 00 1F F8 FF FB E8 0F FF F0 0B 6F FC 07 B3 :0E
C110: CC 05 9F FC 00 3F 3C 00 2F FC 00 33 CE 00 1F FE :30
C120: 01 FF F3 01 AD 7D 01 FF FF 01 9A EB 00 7E 7F 00 :A0
C130: 6D F5 00 1F FE 00 13 AE 00 00 00 00 00 00 00 :40
C140: 00 00 00 00 00 00 07 C7 E0 07 C7 E0 0E 7F F0 0F :E8
C150: EF B0 1F FF F0 1B DD F0 0F FF F0 3F F6 B0 33 CF :7A
C160: E0 3F F9 E0 3C FC 00 3F FC 00 73 CC 00 7F F8 00 :21
C170: CF FF C0 BA F4 C0 FF F8 00 D5 DF FF FE 7F C0 AF :92
C180: F3 C0 7F FC 00 53 CC 00 00 00 00 00 00 00 00 :40
C190: 00 00 00 00 00 00 7F F8 00 77 C8 00 FE 7E 00 AF :E1
C1A0: F6 00 FF FF 8D DB D9 80 CF FF 80 BF F5 80 73 CC :69
C1B0: 00 7F F8 00 3C FC 00 3F FC 00 33 CF E0 3F F9 E0 :E4
C1C0: 0F FF F0 3A F6 00 1F F8 00 15 DF FF 0E 7F F0 0B :90
C1D0: EA D0 07 FF E0 07 C7 E0 00 00 00 00 00 00 00 :4E
C1E0: 00 00 00 00 00 00 60 1F F0 00 1F F0 60 FF FE 90 DB :46
C1F0: FE F0 7F C0 98 6F 40 F8 7F FF 98 57 18 78 7F FF :E7
sum : AF D5 56 C8 E0 62 97 FA 72 80 75 57 37 D9 EB 7E :B9

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C200: FC 3E 7E 7C 7F 99 FE 7D FF BE 47 FF E2 7A BF DE :C3
C210: 7F E7 FE 7B 7F DE 3F 99 FE 29 FF 94 00 7E 7E 01 :CB
C220: AB AE 00 01 FC 00 01 5C 00 00 00 00 00 00 00 :B3
C230: 00 00 00 00 00 60 1F F0 00 1F F0 60 FF FF 9C E0 :65
C240: FE FC 1E 03 9E 1E 02 FE 3F FF 9E 3C 19 76 3F FF :BC
C250: FE 2A 7E 7C 7F 99 FE 7D FF BE 47 FF E2 7A BF DE :B1
C260: 7F E7 FE 53 7F DE 7F 99 F8 3D FF A8 7E 7E 00 55 :59
C270: AB 80 3F 80 00 3A 80 00 00 00 00 00 00 00 00 :A4
C280: 00 00 00 00 00 00 00 00 00 00 00 33 EE 03 FE 7F 03 :E1
C290: CF F5 00 1F FF FF FB EB 03 FF F3 03 2F FD 00 33 :D0
C2A0: CE 1C 1F FE 3E 3F 3C 3E 2F FC 3F F3 CC 3E 9F FC :00
C2B0: 3F FF F0 2F AD 7C 3F FF F8 2E FA E8 1F FE 7D 15 :6E
C2C0: 5F F0 0F FF E0 0C B3 E0 00 00 00 00 00 00 00 :DC
C2D0: 00 00 00 00 00 00 00 07 FF E0 07 DB E0 0F FE 7D 0B :30
C2E0: F7 F0 00 1F F8 FF FB E8 0F FF F0 00 6F FC 1F B3 :28
C2F0: CC 1E 9F FC 3E 3F 3C 3E 2F FC 3F 33 CE 3E 1F FE :42
sum : 4A 6E 12 B0 96 AA C3 E2 85 2B 83 CC 43 D4 2F 01 :A5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C300: 3F FF F3 2F AD 7D 3F FF FF 2D DA EB 1E 7E 7F 16 :EA
C310: 6D F5 0C 1F FE 0C 13 AE 00 00 00 00 00 00 00 :58
C320: 00 00 00 00 00 00 07 C7 F0 07 C7 B0 0E 7F F8 0F :D0
C330: EA B8 1F FF FC 1B DD 7C 0F FF FC 3F F7 7C 33 CF :EE
C340: FC 3F F9 5C 3C FC 7C 7C FC 6C 73 CC 38 7F B8 38 :11
C350: CF FF F0 BA F4 B0 FF F8 00 D5 DF FF FE 7F C0 AF :B2
C360: F3 C0 7F FC 00 53 CC 00 00 00 00 00 00 00 00 :40
C370: 00 00 00 00 00 00 7F F8 30 77 C8 30 FE 7E 38 AF :79
C380: F6 38 FF FF FC DB D9 7C CF FF FC BF F7 7C 73 CD :94
C390: FC 7F F9 5C 3C FC 7C 3F FC 6C 33 CF F8 3F F9 B8 :15
C3A0: 0F FF F0 3A F6 B0 1F F8 00 15 DF FF 0E 7F F0 0B :70
C3B0: EA D0 07 FF E0 07 C7 E0 00 00 00 00 00 00 00 :4E
C3C0: 00 00 00 00 00 00 60 1F F0 98 00 00 60 FF E3 9C 3F :55
C3D0: 80 F0 FF FF 9E 63 40 F8 FF FF 9E 51 18 78 FF FF :22
C3E0: FF 2A 7E 54 FF 99 FF 41 FF 82 E7 FF E7 5A BF DA :14
C3F0: FF E7 FF 43 7F C2 FF 99 FF 29 7F 94 7E 7E 7F 01 :B8
sum : B0 31 F1 89 01 4F F4 45 8A 15 C9 A6 D0 62 CF 33 :33

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: AB AA 03 FF FF 00 01 5C 00 00 00 00 00 00 00 :B3
C410: 00 00 00 00 00 60 00 03 9E 00 00 60 3F 07 9F 00 :46
C420: 01 FC 7F FF 9F 1E 02 F2 FF FF 9F 24 19 72 FF FF :76
C430: FF 2A 7E 6C FF 99 FF 41 FF 82 E7 FF E7 5A BF DA :2C
C440: FF E7 FF 43 7F C2 FF 99 FF 29 FF AC FE 7E 7C 55 :21
C450: AB 80 FF FF C0 2A 80 00 00 00 00 00 00 00 00 :93
C460: 00 7F FF 00 00 00 07 FF FF 00 33 EE 0F FE 7F 03 :33
C470: CD 05 00 1F FF FF FB E9 0F FF F3 03 2F FD 07 F3 :CD
C480: CF 00 1F FE 0F FF 3F 00 2F FC 1F F3 CF 07 9F FC :E7
C490: 3F FF F3 09 6D 7C 3F FF FE 0A 1B A8 3F FE 7C 09 :EE
C4A0: 15 90 1F FF F8 06 B2 A0 00 00 00 00 00 00 00 :13
C4B0: 0F FF F0 00 00 00 1F FF F8 07 DB E0 3F FE 7C 08 :97
C4C0: F5 90 00 1F FE FF FB E8 3F FF F3 0B 6F FC 1F F3 :30
C4D0: CF 05 9F FC 0F FF 3F 00 2F FC 03 F3 CF 00 1F FE :C9
C4E0: 07 FF F3 01 AD 7D 07 FF FF 01 92 E9 03 FE 7F 00 :25
C4F0: 4D 91 00 FF FF 00 33 AE 00 00 00 00 00 00 00 :BD
sum : 6C 3E B0 EC 08 FE 46 46 3B B2 48 82 09 49 B3 22 :B6

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C500: 0F FF E0 00 00 00 1F FF F8 05 4D 60 3E 7F FC 09 :78
C510: A8 90 7F FF FC 1B D8 50 CF FF FC 3F F6 90 F3 CF :46
C520: F8 3F F9 E0 FC FF F0 3F FC 00 F3 CF E0 7F F8 00 :4F
C530: CF FF F0 BA F4 C0 FF F8 00 95 DF FF FE 7F F0 A9 :AC
C540: B3 C0 FF FF E0 77 CC 00 00 00 00 00 00 00 00 :94
C550: FF FE 00 00 00 00 FF FF 00 75 CC 00 FE 7F C0 89 :02
C560: B2 00 FF FF E0 97 C9 80 CF FF E0 BF F5 80 F3 CF :14
C570: C0 7F F8 00 FC FF F0 3F FC 00 F3 CF F8 3F F9 A0 :EF
C580: CF FF FC 3A F6 D0 7F F8 00 15 DF FF 3E 7F FC 09 :F6
C590: AF 10 1F FF F8 07 DB E0 00 00 00 00 00 00 00 :97
C5A0: 3F F8 00 00 00 60 FF FF 98 1F F0 60 FF FF 9C ED :23
C5B0: FE F0 FF FF 9E 23 40 F8 FF FF 9E 51 18 78 FF FF :60
C5C0: FF 2A 7E 54 FF 99 FF 41 FF 82 E7 FF E7 5A BF DA :14
C5D0: FF E7 FF 43 7F C2 FF 99 FF 29 FF 94 7E 7E 7F 01 :38
C5E0: AB AA 03 FF FF 00 01 5C 00 00 00 00 00 00 00 :B3
C5F0: 3F F8 00 00 60 FF FF 9E 1F F0 60 FF FF 9F ED :2C
sum : 45 B4 D8 65 B1 FC 01 48 C1 0A FD 9E B6 18 F7 36 :8D

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C600: FE FC FF FF 9F 1E 02 F2 FF FF 9F 24 19 72 FF FF :F3
C610: FF 2A 7E 6C FF 99 FF 41 FF 82 E7 FF E7 5A BF DA :2C
C620: FF E7 FF 43 7F C2 FF 99 FF 29 FF AC FE 7E 7C 55 :21
C630: AB 80 FF FF C0 2A 80 00 00 00 00 00 00 00 00 :93
C640: 00 7F FF 00 00 00 07 FF FF 00 33 EE 0F FE 7F 03 :33
C650: CD 05 00 1F FF FF FB E9 3F FF F3 0D 2F FD 7F F3 :7F
C660: CF 1C 1F FE FF FF 3F 3E 2F FC FF F3 CF 3E 9F FC :48
C670: FF FF F3 2E ED 7C FF FF FE 2E 1A E8 7F FE 7C 15 :C2
C680: 15 90 3F FF F8 0C B2 A0 00 00 00 00 00 00 00 :39
C690: 0F FF F0 00 00 00 1F FF F8 07 DB E0 3F FE 7C 08 :97
C6A0: F5 90 00 1F FE FF FB E8 3F FF F3 0D 6F FC 7F F3 :9F
C6B0: CF 1D 9F FC FF FF 3F 3E 2F FC FF F3 CF 3E 9F FE :C9
C6C0: FF FF F3 2E AD 7D FF FF FF 2E 92 E9 7F FE 7F 14 :FF
C6D0: 4D 91 3F FF FF 0C 33 AE 00 00 00 00 00 00 00 :08
C6E0: 0F FF F8 00 00 00 1F FF FC 05 4D 30 3E 7F FE 09 :66
C6F0: AB B8 7F FF FF 1B D8 7C CF FF FF 3F F7 7C F3 CF :8D
sum : 2D 7F 03 3E 68 CB F4 DE 98 07 6F DD BB B2 5D 1A :C1

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C700: FF 3F F9 5C FC FF FF 3F FC 64 F3 CF FE 7F F8 38 :9B
C710: CF FF FC BA F4 B0 FF F8 00 95 DF FF FE 7F F0 A9 :A8
C720: B3 C0 FF FF E0 77 CC 00 00 00 00 00 00 00 00 :94
C730: FF FE 78 00 00 00 FF FF FC 75 CC 30 FE 7F FE 89 :E4
C740: B2 38 FF FF FF 97 C9 7C CF FF FF BF F5 7C F3 CF :82
C750: FF 7F F9 5C FC FF 3F 3F FC 64 F3 CF FE 7F F9 B8 :1C
C760: CF FF FC 3A F6 B0 7F F8 00 15 DF FF 3E 7F FC 09 :D6
C770: AF 10 1F FF F8 07 DB E0 00 00 00 00 00 00 00 :97
C780: 00 18 00 00 18 60 00 7E 00 00 7E 60 01 99 90 01 :17
C790: FF B0 3F FF FC 3A BF DC 47 E7 E2 7B 7F DE 7F 99 :BE
C7A0: FE 69 FF 9E 7E 7E 7E 55 D5 D6 3F FF FC 2A AA AC :38
C7B0: 31 E7 8C 3F 43 74 60 C3 06 5F 42 FA 00 00 00 3F :9D
C7C0: 81 FC 00 00 00 DB 66 DB 00 00 00 00 00 00 00 :99
C7D0: 00 18 00 00 18 60 00 7E 00 00 7E 60 01 99 90 01 :17
C7E0: FF B0 3F FF FC 3A BF DC 47 E7 E2 7B 7F DE 7F 99 :BE
C7F0: FE 69 FF 9E 7E 7E 7E 55 D5 D6 3F FF FC 2A AA AC :38
sum : 5B 07 87 22 20 F2 2B C5 01 BF EF 39 23 F9 40 C5 :16

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: 31 E7 8C 3F 43 74 60 C3 06 5F 42 FA 35 81 AC 2A :EA
C810: 81 54 00 00 00 00 00 00 00 00 00 00 00 00 :D5
C820: 00 00 00 00 00 00 03 FF E0 03 E3 ED 04 FF 80 07 :3F
C830: FD 7A 0F FF 00 FF AF FD 3F CF 80 3F F7 7A 4F 3F :FC
C840: E0 7F EE AD FC FF 00 BF F7 00 4F 3F E0 7F EE ED :73
C850: 3F CF 80 35 F7 FA 0F FF 00 0B AE FD 04 FF 80 07 :02
C860: F5 7A 03 FF E0 03 A2 AD 00 00 00 00 00 00 00 :A3
C870: 00 00 00 00 00 03 FF E0 03 E3 ED 04 FF 90 07 :42
C880: FD 78 1F FF 00 EF AF F8 3F CF 90 3F F7 78 4F 3F :03
C890: E0 7F EE AD FC FF 00 BF F7 00 4F 3F E0 7F EE ED :59
C8A0: 3F CF 90 35 F7 F8 0F FF 00 0B AE F8 04 FF 90 07 :18
C8B0: F5 78 03 FF E0 03 A2 AD 00 00 00 00 00 00 00 :94
C8C0: 00 00 00 00 00 07 FF C0 B7 C7 C0 01 FF 20 5F :83
C8D0: BF E0 00 FF F0 BF 75 F0 01 F3 FC 5F AF FC 07 FC :AF
C8E0: F2 B7 57 FE 00 FF 3F 00 AF FF 07 FC F2 B5 57 FE :E9
C8F0: 01 F3 FC 5E AF AC 00 FF F0 BF F5 DF 01 FF 20 5E :A9
sum : 86 45 FF 4D 88 C2 E1 6D 92 81 D1 B2 96 1C E4 48 :23

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: BF E0 07 FF C0 B5 45 C0 00 00 00 00 00 00 00 :1F
C910: 00 00 00 00 00 00 07 FF C0 07 C7 C0 09 FF 20 1F :98
C920: BF E0 00 FF F0 1F 75 F0 09 F3 FC 1F AF FC 07 FC :D7
C930: F2 07 57 FE 00 FF 3F 00 AF FF 07 FC F2 05 57 FE :89
C940: 09 F3 FC 1E AF AC 00 FF F0 1F F5 DF 09 FF 20 1E :99
C950: BF E0 07 FF C0 05 45 C0 00 00 00 00 00 00 00 :6F
C960: 3F 18 00 3F 18 60 7F FE 00 40 7E 60 FF 99 90 81 :52
C970: FF B0 FF FF FC BA BF DC 47 E7 E2 7B 7F DE 7F 99 :FE
C980: FE 69 FF 9E 7E 7E 7E 55 D5 D6 3F FF FC 2A AA AC :38
C990: 31 E7 8C 3F 43 74 60 C3 06 5F 42 FA 00 00 00 3F :9D
C9A0: 81 FC 00 00 00 DB 66 DB 00 00 00 00 00 00 00 :99
C9B0: 3F 18 00 3F 18 60 7F FE 00 40 7E 60 FF 99 90 81 :52
C9C0: FF B0 FF FF FC BA BF DC 47 E7 E2 7B 7F DE 7F 99 :FE
C9D0: FE 69 FF 9E 7E 7E 7E 55 D5 D6 3F FF FC 2A AA AC :38
C9E0: 31 E7 8C 3F 43 74 60 C3 06 5F 42 FA 35 81 AC 2A :EA
C9F0: 81 54 00 00 00 00 00 00 00 00 00 00 00 00 :D5
sum : 14 1A 75 4F C9 77 E3 2D AC D0 81 62 DC C2 BC 2C :27

```



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CA00:	00	00	00	00	00	00	03	FF	E0	03	E3	ED	04	FF	80	07	:3F
CA10:	FD	7A	0F	FF	00	FF	AF	FD	3F	CF	80	3F	F7	7A	4F	3F	:FC
CA20:	E0	7F	EE	AD	FC	FF	00	BF	F7	00	CF	3F	ED	07	EE	ED	:F3
CA30:	FF	CF	80	B5	F7	FA	FF	FF	00	8F	AE	FD	FC	FF	80	D7	:7E
CA40:	F5	7A	7F	FF	E0	73	A2	AD	00	00	00	00	00	00	00	00	:8F
CA50:	00	00	00	00	00	00	03	FF	E0	03	E3	E0	04	FF	90	07	:42
CA60:	FD	78	0F	FF	00	FF	AF	F8	3F	CF	90	3F	F7	78	4F	3F	:03
CA70:	E0	7F	EE	A0	FC	FF	00	BF	F7	00	CF	3F	E0	7F	EE	E0	:09
CA80:	FF	CF	90	B5	F7	F8	FF	FF	00	8F	AE	F8	FC	FF	90	D7	:97
CA90:	F5	78	7F	FF	E0	73	A2	A0	00	00	00	00	00	00	00	00	:80
CAA0:	00	00	78	00	00	78	07	FF	FE	B7	C7	D6	01	FF	3F	5F	:E6
CAB0:	BF	EB	00	FF	FF	BF	75	F3	01	F3	FF	5F	AF	FD	07	FC	:D0
CAC0:	F3	B7	57	FE	00	FF	3F	00	AF	FF	07	FC	F2	B5	57	FE	:EA
CAD0:	01	F3	FC	5E	AF	AC	00	FF	F0	BF	F5	DF	01	FF	20	5E	:A9
CAE0:	BF	E0	07	FF	C0	B5	45	C0	00	00	00	00	00	00	00	00	:1F
CAF0:	00	00	78	00	00	78	07	FF	FE	07	C7	D6	09	FF	3F	1F	:FE
sum :	14	F5	52	0D	14	E3	AD	6C	C8	31	59	A4	5A	9B	96	DD	:D6

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CB00:	BF	EB	00	FF	FF	1F	75	F3	09	F3	FF	1F	AF	FD	07	FC	:F8
CB10:	F3	07	57	FE	00	FF	3F	00	AF	FF	07	FC	F2	05	57	FE	:8A
CB20:	09	F3	FC	1E	AF	AC	00	FF	F0	1F	F5	DF	09	FF	20	1E	:99
CB30:	BF	E0	07	FF	C0	05	45	C0	00	00	00	00	00	00	00	00	:6F
CB40:	00	FF	00	00	18	60	03	FF	98	00	7E	60	7F	99	DE	01	:E6
CB50:	FF	B0	FF	FF	FF	2A	BF	D4	C7	E7	E3	78	7F	DE	FF	99	:6A
CB60:	FF	41	FF	82	FE	7E	7F	55	D5	D6	FF	FF	FF	2A	AA	AC	:39
CB70:	F1	FF	8F	3F	43	74	E0	FF	07	5F	42	FA	40	66	02	3F	:D0
CB80:	81	FC	00	00	00	DB	66	DB	00	00	00	00	00	00	00	00	:99
CB90:	00	FF	00	00	18	60	03	FF	98	00	7E	60	7F	99	DE	01	:E6
CBA0:	FF	B0	FF	FF	FF	2A	BF	D4	C7	E7	E3	78	7F	DE	FF	99	:6A
CB80:	FF	41	FF	82	FE	7E	7F	55	D5	D6	FF	FF	FF	2A	AA	AC	:39
CBC0:	F1	FF	8F	3F	43	74	E0	FF	07	5F	42	FA	F5	E7	AF	2A	:AB
CBD0:	81	54	7F	C3	FE	00	00	00	00	00	00	00	00	00	00	00	:15
CBE0:	07	FF	00	00	00	00	0F	FF	F0	02	E3	AD	1C	FF	80	07	:28
CBF0:	5D	7A	0F	FF	00	FF	A2	FD	FF	CF	80	3F	F0	7A	CF	3F	:88
sum :	BE	6C	F2	5C	1C	A1	52	D7	0D	1A	A2	8E	E5	09	8C	53	:82

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CC00:	F0	7F	E8	AD	FC	FF	F0	BF	F1	00	CF	3F	F0	7F	E8	AD	:B1
CC10:	FF	CF	80	35	F0	7A	7F	FF	00	0B	A2	FD	1C	FF	80	07	:B7
CC20:	DD	7A	0F	FF	F0	02	E3	AD	00	00	00	00	00	00	00	00	:E7
CC30:	07	FF	F0	00	00	00	0F	FF	FC	02	E3	A0	1C	FF	96	07	:3D
CC40:	5D	78	0F	FF	06	FF	A2	F8	FF	CF	96	3F	F0	78	CF	3F	:9B
CC50:	FC	7F	E8	AD	FC	FF	F0	BF	F1	00	CF	3F	FC	F0	E8	AD	:AF
CC60:	FF	CF	96	35	F0	78	7F	FF	06	0B	A2	F8	1C	FF	96	07	:E2
CC70:	DD	78	0F	FF	FC	02	E3	A0	00	00	00	00	00	00	00	00	:E4
CC80:	0F	FF	E0	00	00	0F	FF	F0	B5	C7	C0	01	FF	38	5E	:8E	
CC90:	BF	E0	00	FF	FE	BF	45	F0	01	F3	FF	5E	0F	FC	0F	FC	:F3
CCA0:	F3	B5	17	FE	0F	FF	3F	00	8F	FF	0F	FC	F3	B5	17	FE	:60
CCB0:	01	F3	FF	5E	0F	AC	00	FF	F0	BF	45	DF	01	FF	38	5E	:74
CCC0:	BA	E0	0F	FF	F0	B5	C5	C0	00	00	00	00	00	00	00	00	:D2
CCD0:	0F	FF	E0	00	00	00	3F	FF	F0	05	C7	C0	69	FF	38	1E	:66
CCE0:	BB	E0	60	FF	FE	1F	45	F0	69	F3	FF	1E	0F	FC	3F	FC	:0B
CCF0:	F3	05	17	FE	0F	FF	3F	00	8F	FF	3F	FC	F3	05	17	FE	:30
sum :	3D	50	5F	0B	E3	30	70	5D	3B	44	7A	25	9F	22	6F	6F	:94

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CD00:	69	F3	FF	1E	0F	AC	60	FF	F0	1F	45	DF	69	FF	38	1E	:84
CD10:	BA	E0	3F	FF	F0	05	C5	C0	00	00	00	00	00	00	00	00	:52
CD20:	FF	FF	00	3F	18	60	FF	FF	98	40	7E	60	FF	99	DE	81	:60
CD30:	FF	B0	FF	FF	FF	AA	BF	D4	C7	E7	E3	78	7F	DE	FF	99	:EA
CD40:	FF	41	FF	82	FE	7E	7F	55	D5	D6	FF	FF	FF	2A	AA	AC	:39
CD50:	F1	FF	8F	3F	43	74	E0	FF	07	5F	42	FA	40	66	02	3F	:D0
CD60:	81	FC	00	00	00	DB	66	DB	00	00	00	00	00	00	00	00	:99
CD70:	FF	FF	00	00	18	60	FF	FF	98	40	7E	60	FF	99	DE	81	:60
CD80:	FF	B0	FF	FF	FF	AA	BF	D4	C7	E7	E3	78	7F	DE	FF	99	:EA
CD90:	FF	41	FF	82	FE	7E	7F	55	D5	D6	FF	FF	FF	2A	AA	AC	:39
CDA0:	F1	FF	8F	3F	43	74	E0	FF	07	5F	42	FA	F5	E7	AF	2A	:AB
CDB0:	81	54	7F	C3	FE	00	00	00	00	00	00	00	00	00	00	00	:15
CDC0:	07	FF	F0	00	00	00	0F	FF	F0	02	E3	AD	1C	FF	80	07	:28
CDD0:	5D	7A	0F	FF	00	FF	A2	FD	FF	CF	80	3F	F0	7A	CF	3F	:88
CDE0:	F0	7F	E8	AD	FC	FF	F0	BF	F1	00	CF	3F	F0	7F	E8	AD	:B1
CDF0:	FF	CF	80	B5	F0	7A	FF	FF	00	8B	A2	FD	FC	FF	80	C7	:07
sum :	54	C8	3E	3F	99	FC	65	A2	46	33	5D	AF	90	85	AE	CD	:4A

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CE00:	DD	7A	7F	FF	F0	72	E3	AD	00	00	00	00	00	00	00	00	:C7
CE10:	07	FF	F0	00	00	00	0F	FF	FC	02	E3	A0	1C	FF	96	07	:3D
CE20:	5D	78	0F	FF	06	FF	A2	F8	FF	CF	96	3F	F0	78	CF	3F	:9B
CE30:	FC	7F	E8	AD	FC	FF	F0	BF	F1	00	CF	3F	FC	F0	E8	AD	:AF
CE40:	FF	CF	96	B5	F0	78	FF	FF	06	8B	A2	F8	FC	FF	96	C7	:02
CE50:	DD	78	7F	FF	FC	72	E3	A0	00	00	00	00	00	00	00	00	:C4
CE60:	0F	FF	FF	00	00	00	0F	FF	FF	B5	C7	C0	01	FF	3F	5E	:01
CE70:	BB	E3	00	FF	FF	BF	45	F1	01	F3	FF	5E	0F	FC	0F	FC	:F9
CE80:	F3	B5	17	FE	0F	FF	3F	00	8F	FF	0F	FC	F3	B5	17	FE	:60
CE90:	01	F3	FF	5E	0F	AC	00	FF	F0	BF	45	DF	01	FF	38	5E	:74
CEA0:	BA	E0	0F	FF	F0	B5	C5	C0	00	00	00	00	00	00	00	00	:D2
CEB0:	0F	FF	FF	00	00	00	3F	FF	FF	05	C7	C0	69	FF	3F	1E	:A9
CEC0:	BB	E3	60	FF	FF	1F	45	F1	69	F3	FF	1E	0F	FC	3F	FC	:11
CED0:	F3	05	17	FE	0F	FF	3F	00	8F	FF	3F	FC	F3	05	17	FE	:30
CEE0:	69	F3	FF	1E	0F	AC	60	FF	F0	1F	45	DF	69	FF	38	1E	:84
CEF0:	BA	E0	3F	FF	F0	05	C5	C0	00	00	00	00	00	00	00	00	:52
sum :	71	DB	53	C6	F8	48	A6	60	58	DB	4E	E4	DC	A5	4D	99	:74

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
CF00:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
CF10:	5A	00	00	00	00	03	7E	C0	00	00	00	05	FF	A0	00	00	:3F
CF20:	00	08	FF	00	00	00	00	08	FF	00	00	00	00	05	FF	A0	:58
CF30:	00	00	00	03	7F	40	00										



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D400: 00 00 00 80 00 00 00 00 00 90 00 00 00 00 00 00 C6 :D6
D410: 01 30 00 00 00 E0 1B E0 00 03 00 F4 CF C0 80 0C :1E
D420: 00 FB 7F 08 A0 80 00 FF FE C0 85 D0 00 FF F9 00 :0C
D430: 07 40 00 FF F6 00 DD 00 00 FF F9 98 F4 80 00 FF :EC
D440: FF C0 FD 54 00 FF FF FF 00 00 00 00 00 00 00 :0D
D450: FD 54 00 FF FF FF F4 80 00 FF FF C0 DD 00 00 FF :5C
D460: F9 98 D7 40 00 FF F6 00 85 D0 00 FF F9 00 A0 80 :3A
D470: 00 FF FE C0 80 0C 00 FB 7F 08 00 03 00 F4 CF C0 :51
D480: 00 00 00 E0 1B E0 00 00 00 C6 01 30 00 00 00 90 :62
D490: 00 00 00 00 00 80 00 00 00 00 00 00 00 00 00 :80
D4A0: 3F 80 00 3A 80 00 7E 7E 00 75 D5 80 7F 99 FC 29 :7C
D4B0: FF 94 7F E7 FE 7B 7F DE 47 FF E2 7A BF DE 7F 99 :26
D4C0: FE 7D FF BE 3F FF FE 3E 7E 7C 19 FF FE 1E 18 EA :E2
D4D0: 19 03 FE 1F 02 F6 09 01 FC 0F 01 FC 00 00 00 06 :49
D4E0: 00 00 00 00 00 06 00 00 00 00 00 00 00 00 00 :06
D4F0: 3F 80 00 3A 80 00 7E 7E 00 75 D5 80 7F 99 FC 29 :7C
sum : 61 2A CD F2 6F 6F 63 72 C3 63 24 C3 54 61 77 AB :E1

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D500: FF 94 7F E7 FE 7B 7F DE 47 FF E2 7A BF DE 7F 99 :26
D510: FE 7D FF BE 3F FF FE 3E 7E 7C 19 FF FE 1E 18 EA :E2
D520: 19 03 FE 1F 02 F6 09 7F FF 0F 7F DB 00 0F F8 06 :2E
D530: 0F F8 00 00 00 06 00 00 00 00 00 00 00 00 00 :0D
D540: 00 7E 00 00 7E 00 00 FF 00 00 FF 00 00 00 00 :F9
D550: AB 00 1F FF F8 1F 7E F8 3B FF DC 3B 7E DC 71 FF :71
D560: 8E 71 54 8E 73 FF CE 73 AB CE 07 C3 E0 07 C3 E0 :61
D570: 1F C3 F8 18 C3 18 3F C3 FC 20 42 04 3F C3 FC 20 :4F
D580: 42 04 1F 81 F8 1F 81 F8 00 00 00 00 00 00 00 :76
D590: 00 00 00 00 00 00 00 3C 00 00 3C 00 00 7E 00 :F6
D5A0: 6A 00 07 FF E0 07 DB E0 0D FF B0 0D DB B0 18 7E :FC
D5B0: 18 18 56 18 18 FF 18 18 EB 18 01 E7 80 01 E7 80 :B8
D5C0: 03 C3 C0 03 C3 C0 07 C3 E0 04 42 20 0F C3 F0 08 :E6
D5D0: 42 10 07 81 E0 07 81 E0 00 00 00 00 00 00 00 :22
D5E0: 00 00 00 00 00 03 00 60 03 00 60 03 1C 60 03 :48
D5F0: 1C 60 01 FF C0 01 EB C0 00 7F 80 00 6B 80 00 3E :10
sum : A2 0D 2B 84 3E 99 FB 57 DE 14 4D CA 32 3E 0E CF :DD

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D600: 00 00 3E 00 00 7F 00 00 6B 00 00 E3 80 00 E3 80 :EE
D610: 01 E3 C0 01 E3 C0 03 E3 E0 02 22 20 03 C1 E0 03 :F9
D620: C1 E0 00 00 00 00 00 00 00 00 00 00 00 00 :A1
D630: 00 00 00 00 00 00 00 00 00 00 00 01 00 80 01 :82
D640: 00 80 00 99 00 00 99 00 00 7E 00 00 6E 00 00 3C :DA
D650: 00 00 3C 00 00 3C 00 00 28 00 00 66 00 00 66 :6C
D660: 00 C3 00 00 C3 00 01 C3 80 01 C3 80 00 00 00 :0E
D670: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D680: 00 00 00 00 00 00 FF FF FF D5 55 55 FF FF FF :23
D690: AA AA FF FF FF 95 55 55 FF FF FF FF FF FF :18
D6A0: FF 87 FF FF FF FF 87 FF FF FF FF FF FF FF :90
D6B0: FF FF FF 9A AA AA FF FF FF 85 55 55 FF FF FF :2E
D6C0: AA AA 00 00 00 00 00 00 00 00 00 00 00 00 :54
D6D0: 00 00 00 00 00 00 FF FF FF 55 55 57 FF FF FF :A5
D6E0: AA AF FF FF FF 55 55 5F FF FF FF FF FF FF :57
D6F0: FF FF FF DF FF FF FF FF FF DF FF FF FF DF :90
sum : BD 8E 35 10 4C 0D 42 DD EC 3C E0 76 EB 4A A2 DA :37

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D700: FF FF FF AA AA AF FF FF FF 55 55 57 FF FF FF :A5
D710: AA AB 00 00 00 00 00 00 00 00 00 00 00 00 :55
D720: 00 03 0C 00 03 0C 0F FF FF 0F CB 6D 7F FF FF :6D
D730: 6B 6D 0F FF FF 08 3B 6D FF FF FF FF 98 61 0C 3F :D5
D740: FF 0B 0F FF 0C 3F 0B D5 55 FF FF FF FF 98 61 :5C
D750: 0F FF FF 08 2B 6D 7F FF FF 7E 4B 6D 0F FF FF :7C
D760: 8B 6D 00 03 0C 00 03 0C 00 00 00 00 00 00 :16
D770: 30 C0 00 30 C0 00 FF FF F0 B6 D3 F0 FF FF B6 :F9
D780: D6 7E FF FF F0 B6 DC 10 FF FF FF 86 19 FF FF :7A
D790: 30 FF FB D0 FF FC 30 55 5B D0 FF FF FF 86 19 FF :40
D7A0: FF FF F0 B6 D4 10 FF FF FE B6 D2 7E FF FF F0 :2E
D7B0: D1 F0 30 C0 00 30 C0 00 00 00 00 00 00 00 :A1
D7C0: 00 00 04 03 80 20 0C 60 10 10 18 16 27 88 19 :C8
D7D0: 60 60 30 99 81 01 06 42 01 00 3C 07 E0 10 08 18 :A7
D7E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D7F0: 10 20 00 08 43 80 0C 7C 70 14 80 10 07 88 10 :3E
sum : 23 3D 82 CC B6 02 B2 02 9A 95 E0 4F 48 00 08 26 :BE

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D800: 43 80 10 1C 0C 10 06 02 18 71 04 07 81 38 00 81 :11
D810: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D820: 00 00 00 00 43 80 08 7C 73 14 84 1C 04 08 10 :92
D830: 00 30 14 00 0C 73 01 E2 98 87 04 07 81 38 00 61 :EA
D840: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D850: 00 00 00 00 43 80 00 44 33 00 84 5C 04 09 A0 :BF
D860: 1C 10 95 E0 00 02 19 C0 04 C7 00 00 81 38 00 61 :61
D870: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D880: 00 00 00 01 C2 00 CC 22 00 3A 21 00 05 90 20 08 :C9
D890: 38 1F 00 07 A9 03 98 40 00 E3 20 1C 81 00 63 86 :6B
D8A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D8B0: 00 0C 00 00 46 00 00 33 00 02 44 C0 05 88 38 :58
D8C0: 10 04 00 60 00 01 98 C0 02 03 00 0C 01 00 02 06 :E7
D8D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D8E0: 00 11 00 00 12 00 08 24 00 04 18 04 03 60 28 :06
D8F0: 80 30 00 60 C8 00 33 00 00 0C 00 00 02 00 00 :1F
sum : 27 60 B9 C4 29 89 5F DD 5C 05 AD 72 1C 31 95 F1 :45

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D900: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D910: 00 D3 00 06 AD A0 15 7F 58 2B FF EA 6F FF F4 B7 :3F
D920: FF FB AF FF F5 57 FF FA 2D FF EA 1B 7F 54 05 56 :4C
D930: B0 00 AD 00 00 00 00 00 00 00 00 00 00 00 :5D
D940: 00 3C 00 00 3C 00 03 E7 C0 03 E7 C0 1F 00 F8 1F :02
D950: 00 F8 00 7E 00 00 7E 00 07 C3 E0 07 C3 E0 3C :84
D960: 3C 3C 00 3C 00 FF 00 00 FF 00 0F C3 F0 0F C3 :36
D970: 7C 00 3E 7C 00 3E 00 FF 00 00 FF 00 0F C3 F0 :43
D980: C3 F0 FC 00 3F FC 00 3F 00 00 00 00 00 00 :29
D990: FC 00 3F FC 00 3F 0F C3 F0 0F C3 F0 0F C3 :F9
D9A0: FF 00 7C 00 3E 7C 00 3E 0F C3 F0 0F C3 F0 :F6
D9B0: 00 00 FF 00 3C 00 3C 3C 00 3C 07 C3 E0 07 C3 :43
D9C0: 00 7E 00 00 7E 00 1F 00 F8 1F 00 F8 03 E7 C0 :07
D9D0: E7 C0 00 3C 00 00 3C 00 00 00 00 00 00 00 :1F
D9E0: C3 00 00 C3 00 00 61 86 18 61 86 18 30 C3 0C :B3
D9F0: C3 0C 18 61 86 18 61 86 18 61 86 18 61 86 :07
sum : 92 78 68 97 9B 03 FD E7 72 DF 84 79 06 2B 7B 6D :F2

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DA00: C3 0C 30 C3 0C 30 C3 0C 30 C3 18 61 86 18 61 :BE
DA10: 18 61 86 18 61 86 30 C3 0C 30 C3 0C 61 86 18 :5C
DA20: 86 18 C3 00 00 C3 00 00 00 00 00 00 00 00 :24
DA30: 00 00 C3 00 00 C3 18 61 86 18 61 86 30 C3 :B3
DA40: C3 0C 61 86 18 61 86 18 61 86 18 61 86 18 :9A
DA50: 30 C3 0C 30 C3 0C 30 C3 0C 30 61 86 18 61 :2B
DA60: 61 86 18 61 86 18 30 C3 0C 30 C3 0C 18 61 :13
DA70: 61 86 00 00 C3 00 00 C3 00 00 00 00 00 00 :6D
DA80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DA90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DAF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 16 60 C1 F2 91 C1 F1 91 3B F1 78 E6 CD 3B 54 53 :36

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DB00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DB90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DBF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DC00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DC90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
DCF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
DD00: 07 E0 18 3C 78 1E 78 1E 78 1E 3C 18 07 E0 00 :38
DD10: 03 C0 0F C0 03 C0 03 C0 03 C0 03 C0 1F F8 00 :85
DD20: 1F F8 78 1E 00 3E 03 F8 1F 80 7E 00 7F FE 00 :80
DD30: 3F FE 00 78 01 E0 07 F8 00 1E 78 1E 1F FE 00 :60
DD40: 01 F8 07 F8 1E 78 78 78 7F FE 00 78 00 78 00 :EB
DD50: 3F FC 38 00 3F F8 00 3E 00 1E 78 3E 3F F8 00 :F3
DD60: 07 F8 1E 00 78 00 7F F8 78 1E 78 1E 1F F8 00 :4F
DD70: 7F FE 78 1E 00 78 01 E0 03 C0 03 C0 03 C0 00 :85
DD80: 0F F0 30 1C 3C 0C 1F F8 70 7E 78 0E 1F F8 00 :35
DD90: 1F F8 78 1E 78 1E 1F FE 00 1E 00 78 1F E0 00 :F5
DDA0: 07 E0 1E 78 78 1E 78 1E 7F FE 78 1E 78 1E :52
DDB0: 7F FE 78 1E 78 1E 7F F8 78 1E 78 1E 7F F8 :8D
DDC0: 07 F8 1E 1E 78 00 78 00 78 00 1E 1E 07 F8 00 :DE
DDD0: 7F E0 78 78 78 1E 78 1E 78 1E 78 78 7F E0 :60
DDE0: 7F FC 78 00 78 00 7F F0 78 00 78 00 7F FC :45
DDF0: 7F FE 78 00 78 00 7F F8 78 00 78 00 78 00 :4C
sum : 66 12 38 0E 05 68 A0 6E DB 4C 16 E2 D7 B8 00 00 :B7

```



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
DE00:	07	FE	1E	00	78	00	78	7E	78	1E	1E	1E	07	FE	00	00	:68
DE10:	78	1E	78	1E	78	1E	7F	FE	78	1E	78	1E	78	1E	00	00	:01
DE20:	1F	F8	03	C0	03	C0	03	C0	03	C0	03	C0	1F	F8	00	00	:FD
DE30:	00	1E	00	1E	00	1E	00	1E	00	1E	78	1E	1F	F8	00	00	:43
DE40:	78	1E	78	78	79	E0	7F	80	7F	E0	79	F8	78	7E	00	00	:A4
DE50:	78	00	78	00	78	00	78	00	78	00	78	00	7F	FC	00	00	:4B
DE60:	78	1E	7E	7E	7F	FE	7F	FE	79	9E	78	1E	78	1E	00	00	:CF
DE70:	78	1E	7E	1E	7F	9E	7F	FE	79	FE	78	7E	78	1E	00	00	:CF
DE80:	1F	F8	78	1E	78	1E	78	1E	78	1E	78	1E	1F	F8	00	00	:1C
DE90:	7F	F8	78	1E	78	1E	78	1E	7F	F8	78	00	78	00	00	00	:A0
DEA0:	1F	F8	78	1E	78	1E	78	1E	79	DE	78	7C	1F	F6	00	00	:39
DEB0:	7F	F8	78	1E	78	1E	78	3E	7F	E0	78	F8	78	3E	00	00	:DE
DEC0:	1F	F0	78	3C	78	00	1F	F8	00	1E	78	1E	1F	F8	00	00	:1D
DED0:	3F	FC	03	C0	03	C0	03	C0	03	C0	03	C0	03	C0	00	00	:CD
DEE0:	78	1E	78	1E	78	1E	78	1E	78	1E	78	1E	1F	F8	00	00	:9B
DEF0:	78	1E	78	1E	78	1E	7C	3E	1F	F8	07	E0	01	80	00	00	:FB
sum :	08	94	00	C0	2D	EC	E5	82	65	5E	CC	1C	14	1E	00	00	:89

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
DF00:	78	1E	78	1E	79	9E	7F	FE	7F	FE	7C	3E	70	0E	00	00	:75
DF10:	78	1E	7C	3E	1F	F8	07	E0	1F	F8	7C	3E	78	1E	00	00	:85
DF20:	1E	1E	1E	1E	1E	1E	07	F8	01	E0	01	E0	01	E0	00	00	:56
DF30:	7F	FE	00	7E	01	F8	07	E0	1F	80	7E	00	7F	FE	00	00	:75
DF40:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
DF50:	00	00	00	00	00	00	00	00	00	00	07	80	07	80	00	00	:0E
DF60:	00	00	00	00	00	00	00	00	0F	00	03	00	0C	00	00	00	:1E
DF70:	00	00	00	00	00	00	00	1F	F8	00	00	00	00	00	00	00	:17
DF80:	00	00	01	80	01	80	1F	F8	01	80	01	80	00	00	00	00	:1B
DF90:	00	00	07	80	07	80	00	00	07	80	07	80	00	00	00	00	:1C
DFA0:	07	F8	07	80	07	80	07	80	07	80	07	80	07	F8	00	00	:A1
DFB0:	1F	E0	01	E0	01	E0	01	E0	01	E0	01	E0	1F	E0	00	00	:63
DFC0:	01	E0	07	80	1E	00	1E	00	1E	00	07	80	01	E0	00	00	:2A
DFD0:	07	80	01	E0	00	78	00	78	00	78	01	E0	07	80	00	00	:38
DFF0:	3C	06	66	18	3C	60	01	80	06	3C	18	66	60	3C	00	00	:39
DF00:	00	06	00	18	00	60	01	80	06	00	18	00	60	00	00	00	:70
sum :	F7	9C	90	E8	21	44	FA	7E	07	6A	C9	02	69	FE	00	00	:8B

## リスト7-8 敵ロボットデータ "DATA 5"

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C000:	1F	80	00	1F	80	00	3F	C0	00	3F	C0	00	3F	FF	F8	37	:A9
C010:	7E	E8	7F	C3	FE	5C	FF	3A	FF	81	FF	B9	FF	9D	FF	99	:A7
C020:	FF	B9	FF	9D	7F	99	FE	5C	FF	3A	09	66	F8	0F	7E	A8	:9B
C030:	09	25	FC	0E	3C	64	09	03	FC	0F	02	A4	00	01	F8	06	:94
C040:	01	F8	00	00	00	06	00	00	00	00	00	00	00	00	00	00	:FF
C050:	00	01	F8	00	01	F8	00	03	FC	00	03	FC	1F	FF	FC	17	:21
C060:	7E	EC	7F	C3	FE	5C	FF	3A	FF	81	FF	B9	FF	9D	FF	99	:AB
C070:	FF	B9	FF	9D	7F	99	FE	5C	FF	3A	19	66	F8	1F	7E	A8	:BB
C080:	39	A4	00	2F	3C	00	39	C0	00	2F	40	00	19	80	00	1F	:68
C090:	80	00	00	00	00	06	00	00	00	00	00	00	00	00	00	00	:86
COA0:	00	00	00	00	00	00	00	1F	80	00	17	80	07	FF	FE	07	:41
COB0:	AA	EE	00	7F	FF	FF	D0	57	07	FF	FF	06	3F	9B	00	E0	:01
COC0:	7E	00	FF	EE	00	1C	38	00	FF	F0	0F	E0	70	0E	FF	E0	:FA
COD0:	1F	FF	F0	1A	BF	90	1F	FF	F0	1B	50	80	0F	0E	FF	E0	:9B
COE0:	B7	60	00	0F	C0	00	0A	C0	00	00	00	00	00	00	00	00	:80
COF0:	00	00	00	00	00	00	00	0F	C0	00	0D	C0	1F	FF	E0	1F	:B9
sum :	DA	D5	DF	B2	71	FD	AC	F6	2A	FD	A7	B4	49	8D	A3	E8	:33

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C400:	C3	E0	03	81	C0	03	81	C0	00	00	00	00	00	00	00	00	:2B
C410:	60	FF	06	60	FF	06	F3	FF	CF	F3	81	CF	FF	FF	FF	B3	:7E
C420:	04	CB	FF	FF	FF	B6	00	6B	F1	FF	8F	F1	C3	8F	67	FF	:15
C430:	E6	67	42	E6	0F	FF	F0	0B	81	80	1F	FF	F8	1C	00	38	:19
C440:	33	FF	CC	30	02	0C	7C	7E	3E	7C	00	3E	38	00	1C	38	:8A
C450:	00	1C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:1C
C460:	00	00	00	00	00	00	00	03	FC	00	03	FC	00	0F	FF	00	:0C
C470:	0E	BF	0F	C1	F0	0F	C1	C0	7F	FF	FC	7F	FB	7C	FF	FF	:8B
C480:	FF	F0	3E	07	0F	FF	FF	00	88	13	FF	FF	FF	F0	3E	07	:0E
C490:	7F	FF	FC	55	F8	FC	0F	C1	F0	0A	C1	C0	00	0F	FF	00	:1F
C4A0:	00	FF	00	03	FC	00	02	BC	00	00	00	00	00	00	00	00	:C9
C4B0:	00	00	00	00	00	00	1C	03	FC	1C	03	FC	3F	8F	FF	FF	:42
C4C0:	8E	BF	11	E1	F0	11	E1	C0	03	FF	FC	00	7B	7C	0F	FF	:E4
C4D0:	FF	00	3E	07	0F	FF	FF	00	88	13	0F	FF	FF	00	3E	07	:3E
C4E0:	03	FF	FC	00	7B	FC	11	E1	F0	11	A1	C0	3F	8F	FF	3E	:04
C4F0:	8D	FF	1C	03	FC	14	02	BC	00	00	00	00	00	00	00	00	:79
sum :	F6	96	C6	01	3B	F4	C0	53	E9	79	9D	F2	E4	CE	08	AB	:EB

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C100:	AA	A0	00	7F	F8	FF	D0	50	1F	FF	F8	1E	3F	98	0F	E0	:DA
C110:	78	0C	FF	E8	00	1C	38	00	FF	F0	00	E0	7E	00	FF	EE	:F9
C120:	00	7F	FF	00	3F	93	00	FF	FF	00	D0	A3	00	FF	FE	00	:BE
C130:	D7	6E	00	7F	80	00	6A	80	00	00	00	00	00	00	00	00	:2E
C140:	00	00	00	00	00	00	03	F0	00	03	50	00	07	FF	F0	06	:42
C150:	ED	F0	0F	FF	F8	00	5A	D8	0F	FF	F8	09	F0	58	0E	07	:9B
C160:	F0	07	FF	70	1C	38	00	0F	FF	00	7E	07	00	77	FF	00	:C3
C170:	FF	FF	E0	D9	FC	60	FF	FE	00	EA	08	FF	7F	FF	E0	75	:D7
C180:	55	E0	01	F8	00	00	A8	00	00	00	00	00	00	00	00	00	:D6
C190:	00	00	00	00	00	00	01	FC	00	01	5C	00	7F	FF	00	76	:4E
C1A0:	E8	00	FF	FF	00	C5	5B	00	FF	FE	00	C9	FC	00	7E	07	:50
C1B0:	00	77	FF	00	1C	38	00	0F	FF	00	1E	07	F0	17	FF	30	:33
C1C0:	1F	FF	F8	19	FC	78	1F	FE	00	0A	0B	FF	07	FF	F8	05	:D7
C1D0:	55	F8	03	F0	00	02	80	00	00	00	00	00	00	00	00	00	:F2
C1E0:	1E	00	00	1E	00	00	3F	7E	0C	3F	7E	0C	7F	FF	FE	55	:9F
C1F0:	81	AA	FF	FF	FF	AB	00	D5	FF	FF	FF	A9	81	95	FF	FE	:62
sum :	28	87	E5	4B	DE	75	E0	00	34	22	9B	34	B2	0D	5B	56	:A7

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C500:	00	00	00	00	00	00	3F	C0	00	3F	C0	00	FF	F0	00	EB	:08
C510:	F0	00	0F	83	F0	03	83	F0	3F	FF	FE	3E	DF	FE	FF	FF	:3D
C520:	FF	E0	7C	0F	FF	FF	F0	C1	10	80	FF	FF	FF	E0	7C	0F	:11
C530:	3F	FF	FE	3F	05	FE	0F	83	F0	03	82	80	FF	F0	00	0F	:D3
C540:	F0	00	3F	C0	00	2B	C0	00	00	00	00	00	00	00	00	00	:DA
C550:	00	00	00	00	00	00	3F	C0	38	3F	C0	38	FF	F1	FC	EB	:45
C560:	F1	FC	0F	87	88	03	87	88	3F	FF	C0	3E	DE	00	FF	FF	:35
C570:	F0	E0	7C	00	FF	FF	F0	C1	10	80	FF	FF	F0	E0	7C	00	:D5
C580:	3F	FF	C0	3F	DE	00	0F	87	88	03	85	88	FF	F1	FC	0F	:14
C590:	F1	7C	3F	C0	38	2B	C0	38	00	00	00	00	00	00	00	00	:C7
C5A0:	00	FF	00	00	FF	00	63	FF	C0	6F	FF	F0	62	00	40	6F	:8F
C5B0:	FF	F0	C3	FF	C0	FF	24	FC	FF	FF	FC	C3	24	C3	FF	FF	:32
C5C0:	FF	C3	24	C3	F3	FF	CF	CF	FF	F3	03	FF	C3	FF	00	FF	:EE
C5D0:	03	FF	C3	3C	0C	3F	03	FF	C2	3C	00	3E	03	FF	C6	0F	:61
C5E0:	00	F6	00	FF	00	00	FF	00	00	00	00	00	00	00	00	00	:F4
C5F0:	00	FF	00	00	FF	00	03	FF	C6	0F	FF	F6	02	00	46	0F	:21
sum :	30	DC	FC	14	1E	95	61	84	94	2E	40	00	F6	41	39	2C	:22



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: 1F FF F8 18 7E 18 1F FF F8 19 BF 98 01 FF 80 00 :CA
C810: 6E 00 00 7E 00 00 18 00 00 00 00 00 00 00 00 :04
C820: 00 00 00 00 00 00 00 00 0C 00 00 0C 00 3C 3F 00 :93
C830: 3C 3F 00 FC FF 00 C0 D7 3F FF FC 0C 0C 0C FF FF :69
C840: FF 3F 3F 3F FF FF FF DF DF C0 FF FF FF 37 37 3F :E1
C850: 3F FF FC 0C 0C 0C 00 FC FF 00 C0 DF 00 3C 3F 00 :73
C860: 3C 37 00 00 0C 00 00 0C 00 00 00 00 00 00 00 :8B
C870: 00 00 00 00 00 00 00 00 0C 00 00 0C 0F 00 3F 0F :75
C880: 00 3F 0F C0 FF 00 C0 D7 3F FF FC 0C 0C 0C FF FF :00
C890: FF 3F 3F 3F FF FF FF DF DF C0 FF FF FF 37 37 3F :E1
C8A0: 3F FF FC 0C 0C 0C 0F C0 FF 00 C0 DF 0F 00 3F 0F :28
C8B0: 00 37 00 00 0C 00 00 0C 00 00 00 00 00 00 00 :4F
C8C0: 00 00 00 00 00 00 00 30 00 30 00 00 FC 3C 00 FC :94
C8D0: 3C 00 FF 3F 00 D7 03 00 3F FF FC 30 30 30 FF FF :1C
C8E0: FF FC FC FC FF FF FF 03 7F 7F FF FF FF FC DC DC :A2
C8F0: 3F FF FC 30 30 30 FF 3F 00 FF 03 00 FC 3C 00 DC :1E
sum : FB 62 74 53 D9 34 F5 81 08 44 33 B3 5C A1 C3 4D :E6

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: 3C 00 30 00 00 30 00 00 00 00 00 00 00 00 00 :9C
C910: 00 00 00 00 00 00 30 00 00 30 00 00 FC 00 F0 FC :48
C920: 00 F0 FF 03 F0 D7 03 00 3F FF FC 30 30 30 FF FF :84
C930: FF FC FC FC FF FF FF 03 7F 7F FF FF FF FC DC DC :A2
C940: 3F FF FC 30 30 30 FF 03 F0 FF 03 00 FC 00 F0 DC :86
C950: 00 F0 30 00 00 30 00 00 00 00 00 00 00 00 00 :50
C960: 3C 7E 3C 3C 7E 3C 0D AB 80 0D FF 80 0D FF 80 0D :D9
C970: 7F 80 0D FF 80 0D 7F 80 0C FF 30 0C 8D 30 0C 7E :E5
C980: 30 0C 5E 30 0C 3C 30 0C 2C 30 06 7E 60 06 7E 60 :72
C990: 01 FF 80 01 FF 80 01 FF 80 01 BF 80 01 FF 80 01 :41
C9A0: CF 80 00 7E 00 00 7E 00 00 00 00 00 00 00 00 :4B
C9B0: 00 00 00 00 00 00 00 00 03 00 00 03 00 3F FF 00 :44
C9C0: 3F FF 00 C0 F8 00 C0 50 3F 03 FC 3F 03 FC FF DF :60
C9D0: F6 FF DF FE FF FF FB DF F7 FF FF DF F6 E7 DD FE :36
C9E0: 3F 03 FC 3F 02 AC 00 C0 F8 00 C0 50 00 3F FF 00 :31
C9F0: 3F FF 00 00 03 00 00 03 00 00 00 00 00 00 00 :44
sum : E8 94 59 16 54 16 27 5E 47 EC AD 5A 4B C1 4F 7C :EB

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CA00: 00 00 00 00 00 00 C0 00 00 C0 00 00 FF FC 00 FF :7A
CA10: FC 00 1F 03 00 0A 03 00 3F C0 FC 3F C0 FC 6F FB :8B
CA20: FF 7F FB FF DF FF FF FF FF 7F 6F FB FF 7F BB 9F :14
CA30: 3F C0 FC 35 40 FC 1F 03 00 0A 03 00 FF FC 00 FF :95
CA40: FC 00 C0 00 00 C0 00 00 00 00 00 00 00 00 00 :7C
CA50: 00 7E 00 00 7E 00 01 FF 80 01 FF 80 01 FF 80 01 :7D
CA60: BF 80 01 FF 80 01 CF 80 06 7E 60 06 7E 60 0C 3C :1F
CA70: 30 0C 2C 30 0C 7E 30 0C 5E 30 0C FF 30 0C BF 30 :22
CA80: 0D FF 80 0D 7F 80 0D FF 80 0D 7E 80 0D AB 80 0D :64
CA90: FF 80 3C 7E 3C 3C 7E 3C 00 00 00 00 00 00 00 :9B
CAA0: 7F 81 FE 7F 81 FE 7F 81 FE 55 81 56 7F FF FE 5E :00
CAB0: 7E FA 7F FF FE 5D BF BA 7F FF FE 4B 5F DA 7F FF :48
CAC0: FE 53 7F DA 7F 99 FE 49 FF BA 7F BD FE 55 BD AA :88
CAD0: 7F 99 FE 40 99 52 7F 99 FE 55 99 AA 00 18 00 00 :07
CAE0: 18 00 00 18 00 00 18 00 00 00 00 00 00 00 00 :48
CAF0: 00 00 00 00 00 00 0F FF FF 0D 55 57 0F FF FF 0A :0D
sum : C3 5F E9 A1 7B 76 4E E4 4B 35 43 CE 64 CE 5E 23 :13

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CB00: FF FD 0F FF FF 05 AA BD 0F FF FF 0A 7F CB 00 9F :75
CB10: F0 00 FF F0 FF FF F0 FF 7F F0 00 9F F0 00 FF F0 :B9
CB20: 0F FF FF 0D 55 D7 0F FF FF 0A 00 AD 0F FF FF 05 :1C
CB30: 55 5D 0F FF FF 0A AA AB 00 00 00 00 00 00 00 :1E
CB40: 00 00 00 00 00 00 FF FF F0 AA AA B0 FF FF F0 5F :3F
CB50: FF 0D FF FF F0 AD 55 D0 FF FF F0 D3 FE 80 0F F9 :06
CB60: 00 0F FF 00 0F FF FF 0B FF FF 0F F9 00 0D FF 00 :38
CB70: FF FF F0 A2 BE 80 FF FF F0 50 05 D0 FF FF F0 AA :A9
CB80: AB D0 FF FF F0 D5 55 50 00 00 00 00 00 00 00 :E3
CB90: 00 18 00 00 18 00 00 18 00 00 18 00 7F 99 FE 55 :CB
CBA0: 99 AA 7F 99 FE 40 99 52 7F BD FE 55 BD AA 7F 99 :92
CBB0: FE 49 FF BA 7F FF FE 53 7F DA 7F FF FE 4B 5F DA :28
CBC0: 7F FF FE 5D BF BA 7F FF FE 5E 7E FA 7F 81 FE 55 :F7
CBD0: 81 56 7F 81 FE 7F 81 FE 00 00 00 00 00 00 00 :D3
CBE0: 00 FF 00 00 FF 00 03 FF C0 03 FF C0 0F C3 F0 03 :47
CBF0: 3C C0 3F 00 FC 3C FF 3C FF C3 FF F0 BD 0F FF FF :29
sum : CF 26 43 CC 4C CA 93 84 26 AC BE A0 FF 66 B5 B5 :30

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CC00: FF E1 00 87 FF FF FF A1 00 87 FF FF FF D0 81 0B :E5
CC10: 3F FF FC 3C 3C 3C 0F FF F0 03 00 C0 03 FF C0 03 :74
CC20: 7F C0 00 FF 00 00 AB 00 00 00 00 00 00 00 00 :E9
CC30: 00 FF 00 00 FF 00 03 FF C0 03 FF C0 0F FF F0 03 :83
CC40: 00 C0 3F FF FC 3C 3C 3C F9 FF FF F6 81 0F E0 7F :8A
CC50: FF FF 80 87 E0 7F FF BF 80 87 F9 FF FF D6 81 0B :82
CC60: 3F FF FC 3C 3C 3C 0F FF F0 03 00 C0 03 FF C0 03 :74
CC70: 7F C0 00 FF 00 00 AB 00 00 00 00 00 00 00 00 :E9
CC80: 00 FF 00 00 FF 00 03 FF C0 03 FF C0 0F FF F0 03 :83
CC90: 00 C0 3F FF FC 3C 3C 3C FF FF FF F0 81 0F FF FF :29
CCA0: FF E1 00 87 FF FF FF A1 00 87 FF C3 FF D0 8D 0B :E5
CCB0: 3F 00 FC 3C FF 3C 0F C3 F0 03 3C C0 03 FF C0 03 :38
CCCO: 7F C0 00 FF 00 00 AB 00 00 00 00 00 00 00 00 :E9
CCD0: 00 FF 00 00 FF 00 03 FF C0 03 FF C0 0F FF F0 03 :83
CCE0: 00 C0 3F FF FC 3C 3C 3C FF FF 9F F0 81 6F FF FE :28
CCF0: 07 E1 01 FF FF FE 07 A1 01 FF FF FF 9F D0 81 6B :E6
sum : 3E 1D 32 42 45 E3 EF 14 88 A3 CC 16 55 CD 2E 1A :71

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CD00: 3F FF FC 3C 3C 3C 0F FF F0 03 00 C0 03 FF C0 03 :74
CD10: 7F C0 00 FF 00 00 AB 00 00 00 00 00 00 00 00 :E9
CD20: 00 18 00 00 18 00 00 7E 00 00 7E 00 19 FF 98 19 :F5
CD30: FF 98 19 FF 98 19 E7 98 07 FF E0 06 81 E0 1F FF :4A
CD40: F8 1B 99 F8 7F FF FE 77 FF FE 00 7E 00 00 66 00 :78
CD50: 7E 7E 7E 7E 66 7E 7F FF FE 5F E7 DE 7E 7E 7E 6A :60
CD60: 7E 6A 00 7E 00 00 18 00 00 00 00 00 00 00 00 :7E
CD70: 00 18 00 00 18 00 00 7E 00 00 7E 00 19 FF 98 19 :F5
CD80: FF 98 19 FF 98 19 FC 98 07 FF E0 06 F8 60 1F FF :56
CD90: F8 1B F9 B8 7F FF FE 77 FF FE 00 7E 00 00 7A 00 :AC
CDA0: 07 DE E0 07 DA E0 0F EF F0 0E EB D0 07 DE E0 05 :07
CDB0: DE A0 00 7E 00 00 18 00 00 00 00 00 00 00 00 :14
CDC0: 00 18 00 00 18 00 00 7E 00 00 7E 00 19 FF 98 19 :F5
CDD0: FF 98 19 FF 98 19 FF 98 07 FF E0 04 FF A0 1F FF :9E
CDE0: F8 1B FF D8 7F FF FE 77 FF FE 00 7E 00 00 4A 00 :A2
CDF0: 00 7E 00 00 7E 00 00 FF 00 00 FB 00 00 7E 00 00 :74
sum : 84 04 36 41 87 E2 54 93 F0 67 E7 F8 4B B6 6D BA :AD

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CE00: 7E 00 00 7E 00 00 18 00 00 00 00 00 00 00 00 :14
CE10: 00 18 00 00 18 00 00 7E 00 00 7E 00 19 FF 98 19 :F5
CE20: FF 98 19 FF 98 19 3F 98 07 FF E0 06 1F E0 1F FF :40
CE30: F8 1D 9F F8 7F FF FE 77 FF FE 00 7E 00 00 5E 00 :78
CE40: 07 7B E0 07 5B E0 0F F7 F0 0B D7 70 07 7B E0 05 :53
CE50: 7B A0 00 7E 00 00 18 00 00 00 00 00 00 00 00 :B1
CE60: 7C 00 3E 70 00 0E 7F 3C FE 70 3C 0E 7F FF FE 70 :97
CE70: 66 0E 27 E7 E4 20 DB 04 03 C3 C0 03 BD C0 03 E7 :55
CE80: C0 00 DB 00 03 E7 C0 00 DB 00 03 C3 C0 03 BD C0 :26
CE90: 00 E7 00 00 DB 00 00 7E 00 00 66 00 00 7E 00 00 :24
CEA0: 7E 00 00 3C 00 00 3C 00 00 00 00 00 00 00 00 :F6
CEB0: 7C 00 3E 70 00 0E 7F 3C FE 70 3C 0E 7F FF FE 70 :97
CEC0: 66 0E 27 FF E4 20 DB 04 03 FF C0 03 BD C0 03 FF :C1
CED0: C0 00 DB 00 03 FF C0 0C DB 30 03 FF C0 1F BD F8 :0A
CEE0: 00 FF 00 0F DB F0 00 7E 00 00 66 00 00 7E 00 00 :3B
CEF0: 7E 00 00 3C 00 00 3C 00 00 00 00 00 00 00 00 :F6
sum : 37 EA 18 47 0E 2A 28 0C AE DA FF D8 37 F6 71 9B :84

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: 7C 00 3E 70 00 0E 7F 3C FE 70 3C 0E 7F FF FE 70 :97
CF10: 66 0E 27 FF E4 20 C3 04 03 FF C0 03 81 C0 03 FF :6D
CF20: C0 00 C3 00 03 FF C0 5C C3 3A 03 FF C0 BF 81 FD :9D
CF30: 00 FF 00 7F C3 FE 00 7E 00 0E 66 7B 00 7E 00 2A :24
CF40: 7E 54 00 3C 00 00 7C 00 00 00 00 00 00 00 00 :4A
CF50: 7C 00 3E 70 00 0E 7F 3C FE 70 3C 0E 7F FF FE 70 :97
CF60: 66 0E 27 FF E4 20 DB 04 03 FF C0 03 BD C0 03 FF :C1
CF70: C0 00 DB 00 03 FF C0 0C DB 30 03 FF C0 1F BD F8 :0A
CF80: 00 FF 00 0F DB F0 00 7E 00 00 66 00 00 7E 00 00 :3B
CF90: 7E 00 00 3C 00 00 3C 00 00 00 00 00 00 00 00 :F6
CFA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CFB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CFC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CFD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CFE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CFF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 40 6E 68 E4 6C 4B 94 E4 A0 26 CA 9B BC 58 40 FD :A2

```

## リスト7-9 敵能力データおよび移動データ "DATA6"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: 40 CE 00 C0 80 C8 80 C2 30 C9 00 C8 60 C6 00 00 :4F
C010: 80 C3 A0 C1 00 00 00 00 00 00 00 00 00 00 00 :A4
C020: 10 20 30 90 00 03 00 00 00 00 00 00 00 00 00 :F3
C030: 08 04 0C 08 10 00 00 00 00 00 00 00 00 00 00 :30
C040: 01 00 00 00 00 00 00 00 00 00 00 00 00 30 31 :62
C050: 11 08 11 08 51 28 51 28 14 08 14 08 54 28 54 28 :54
C060: 10 09 10 08 5F 58 10 08 71 88 72 88 74 88 78 88 :E5
C070: 65 58 6A 58 65 58 6A 58 5F 18 5F 18 5F 18 5F 18 :0A
C080: 9F 09 9F 09 9F 09 9F 09 10 08 10 08 10 08 7F 58 :8F
C090: 3F 0A 9F 09 58 58 00 00 6C 58 69 58 5E 29 58 29 :31
COA0: 66 28 DC 29 63 28 D9 29 96 52 9C 52 93 52 99 52 :C6
COB0: 00 0B 00 0B 00 0B DF 59 1F 88 00 01 1F 80 00 09 :A9
COC0: 00 00 00 00 00 03 00 03 00 03 00 03 00 03 03 :12
COD0: AF 56 A5 54 9F 52 9F 51 00 0B 00 0B 00 0A 00 :09
COE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
COF0: 05 04 01 0C 0F 07 03 00 09 01 00 00 00 00 00 :39
sum : 57 BE 27 34 AD 96 44 29 4E BA FA 31 A7 9E CE E2 :48

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C100: 13 13 13 13 13 13 13 12 12 12 12 11 11 11 21 20 :41
C110: 20 20 29 19 19 1A 0A 0A 0A 0A 0A 9A 99 99 A9 A0 :FC
C120: A0 A0 A1 91 91 92 92 92 93 03 03 03 03 02 02 02 :5E
C130: 02 02 02 02 02 02 02 02 20 20 20 20 20 20 20 :10
C140: 12 12 12 12 12 12 02 02 02 02 11 11 11 11 21 :EA
C150: 21 20 29 19 19 1A 1A 1A 1A 1A 1A 1A 1A 1A 19 :B9
C160: 19 19 19 19 2A 2A 2A 2A 2A 2A 29 29 29 29 29 :56
C170: 29 29 29 29 29 29 29 29 20 20 20 20 20 20 20 :48
C180: 19 19 19 19 19 19 19 10 10 10 11 11 11 11 11 :45
C190: 01 01 91 91 91 91 01 11 11 10 11 10 10 10 20 :FA
C1A0: 20 20 29 29 29 29 29 29 29 29 29 29 29 29 29 :7E
C1B0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :00
C1C0: 1A 1A 1A 1A 1A 1A 0A 0A 0A 0A 19 19 19 19 29 :6A
C1D0: 29 20 21 11 11 12 12 12 12 12 12 12 12 12 11 :51
C1E0: 11 11 11 11 22 22 22 22 22 22 22 22 22 22 22 :DC
C1F0: 22 22 22 22 22 22 22 22 20 20 20 20 20 20 20 :10
sum : 1A 10 BD 7D 9F A3 E3 E9 FD 6C 8B 19 18 17 47 5B :50

```



+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C200:	18	18	18	18	18	18	1A	1A	1A	1A	19	19	19	19	20	:B9
C210:	20	20	21	11	11	12	02	02	02	92	91	91	A1	A0	:94	
C220:	A0	A0	A9	99	99	9A	9A	9A	98	08	08	08	0A	0A	:CE	
C230:	0A	0A	0A	0A	0A	0A	0A	20	20	20	20	20	20	20	:50	
C240:	11	11	11	12	12	12	12	02	02	02	92	92	92	92	:0D	
C250:	02	12	11	11	10	20	29	29	29	29	29	29	29	29	:00	
C260:	29	29	20	20	20	29	29	29	29	29	29	29	29	29	:75	
C270:	29	29	29	29	29	29	29	20	20	20	20	20	20	20	:48	
C280:	21	21	21	21	21	21	21	21	21	20	20	21	21	21	:0E	
C290:	21	11	02	91	A1	A1	A1	A2	A2	A2	A2	92	92	92	:19	
C2A0:	92	92	11	11	22	22	22	22	22	22	22	22	22	22	:DE	
C2B0:	22	22	22	22	22	22	22	20	20	20	20	20	20	20	:10	
C2C0:	11	11	11	11	11	11	10	10	10	19	19	19	19	19	:3D	
C2D0:	09	09	99	99	99	99	09	19	10	19	10	10	10	20	:4A	
C2E0:	20	20	21	21	21	21	21	21	21	21	21	21	21	20	:0D	
C2F0:	20	20	20	20	20	20	20	20	20	20	20	20	20	20	:00	
sum :	9A	9A	9B	0B	2B	46	AF	BD	BA	21	32	48	38	37	67	:AE

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C300:	29	29	29	29	29	29	29	29	29	20	20	29	29	29	:7E	
C310:	29	19	0A	99	A9	A9	A9	AA	AA	AA	AA	9A	9A	9A	:99	
C320:	9A	9A	9A	9A	9A	9A	9A	9A	9A	9A	9A	9A	9A	9A	:A0	
C330:	9A	9A	9A	9A	9A	9A	9A	20	20	20	20	20	20	20	:D0	
C340:	19	19	19	1A	1A	1A	1A	0A	0A	0A	9A	9A	9A	9A	:5D	
C350:	0A	1A	19	19	10	20	21	21	21	21	21	21	21	21	:00	
C360:	21	21	20	20	20	21	21	21	21	21	21	21	21	21	:0D	
C370:	21	21	21	21	21	21	21	20	20	20	20	20	20	20	:08	
C380:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C390:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
C3F0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
sum :	EB	EB	DA	6A	71	82	83	83	F9	F9	F0	80	79	79	E9	:C9

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C000:	01	01	01	01	01	01	0B	03	03	03	03	03	03	03	:2A	
C010:	01	01	01	01	01	01	0B	03	03	03	03	03	03	03	:2A	
C020:	01	01	01	01	01	01	19	11	11	11	11	11	11	11	:A8	
C030:	01	3B	70	01	01	01	01	01	01	01	01	01	01	01	:B9	
C040:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C050:	01	01	68	68	68	68	01	01	68	68	68	68	68	01	:16	
C060:	01	01	02	01	01	01	69	01	01	02	01	01	01	01	:7A	
C070:	01	01	02	01	01	01	01	01	69	68	68	68	68	01	:15	
C080:	01	01	02	01	01	68	68	01	01	01	01	01	02	01	:E0	
C090:	01	01	02	01	01	01	02	01	01	68	01	01	01	02	:7A	
C0A0:	01	01	69	68	68	68	69	01	01	69	68	68	68	01	:1A	
C0B0:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C0C0:	01	01	01	01	DC	DE	01	01	01	01	DC	DE	01	01	:80	
C0D0:	01	01	01	01	DD	DF	01	01	01	01	DD	DF	01	01	:84	
C0E0:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C0F0:	F2	F2	F2	F2	F2	F2	F2	F2	F2	F2	F2	F2	F2	F2	:20	
sum :	01	3B	43	CF	86	F1	A0	2D	15	B4	01	05	4A	4D	15	:22

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C100:	01	01	01	01	01	01	01	01	01	01	3A	30	39	01	:B0	
C110:	01	01	01	01	01	01	4C	01	01	4C	01	07	B1	07	:62	
C120:	01	01	01	01	01	01	01	01	01	01	01	37	30	38	:AC	
C130:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C140:	01	01	01	4A	40	40	40	40	40	40	40	40	40	40	:53	
C150:	01	01	01	42	01	01	01	01	01	01	01	01	08	01	:58	
C160:	01	01	01	42	01	01	01	01	01	01	01	01	08	01	:58	
C170:	01	01	01	42	01	01	A8	A9	AA	AB	01	01	08	01	:FA	
C180:	01	01	01	08	01	01	AB	BC	BE	A8	01	01	08	01	:E7	
C190:	01	01	01	08	01	01	AA	BD	BF	A9	01	01	08	01	:E9	
C1A0:	01	01	01	08	01	01	01	01	01	01	01	01	08	01	:1E	
C1B0:	01	01	01	08	01	B6	B7	01	01	B6	B7	01	08	01	:F4	
C1C0:	01	01	01	08	01	01	01	01	01	01	01	01	08	01	:1E	
C1D0:	01	01	01	08	01	01	01	04	D5	01	01	01	08	01	:C5	
C1E0:	01	01	01	08	01	01	01	01	01	01	01	01	08	01	:1E	
C1F0:	01	01	01	08	01	01	01	01	01	01	01	E3	08	01	:00	
sum :	10	10	10	54	4F	04	4A	41	47	48	05	A6	B0	C4	4F	:AE

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C200:	40	40	40	45	40	40	40	40	40	40	40	48	01	01	:50	
C210:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C220:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C230:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C240:	01	01	01	01	01	01	01	A0	A0	01	01	01	01	01	:4E	
C250:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C260:	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:10	
C270:	68	68	68	68	68	68	68	68	68	68	68	68	68	68	:80	
C280:	02	02	02	02	02	02	02	02	02	02	02	02	02	02	:20	
C290:	02	02	02	02	02	02	02	02	02	02	F8	02	02	02	:16	
C2A0:	02	02	02	02	F9	02	02	02	02	02	02	02	02	02	:17	
C2B0:	02	02	02	02	02	02	02	02	02	02	02	02	02	02	:20	
C2C0:	02	02	02	02	02	02	02	02	02	02	02	02	02	02	:20	
C2D0:	69	69	69	69	69	69	69	69	69	69	69	69	69	69	:90	
C2E0:	05	05	01	01	3C	01	01	05	05	01	01	3C	01	01	:9E	
C2F0:	05	05	01	01	01	01	01	05	05	01	01	01	01	01	:28	
sum :	2B	2B	23	28	55	23	23	CA	CA	23	23	54	2B	E4	EC	:51

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	sum
C300:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C310:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C320:	05	05	01	01	01	98	01	05	05	01	98	01	01	05	:56	
C330:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C340:	05	05	68	68	68	68	05	05	68	68	68	68	68	05	:2E	
C350:	05	05	69	69	69	69	05	05	69	69	69	69	69	05	:38	
C360:	05	05	01	01	01	01	05	05	01	01	01	01	F7	05	:1E	
C370:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C380:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C390:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:28	
C3A0:	05	05	01	01	01	01	05	05	01	01	01	01	01	05	:	



```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C800: F0 01 01 01 01 01 01 01 01 01 01 01 01 01 :FF
C810: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
C820: 01 01 01 01 01 01 01 01 A6 D2 A6 01 01 01 01 :2B
C830: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
C840: 01 4F 70 71 01 01 01 01 01 01 01 01 01 :30
C850: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
C860: 30 30 30 30 30 30 30 30 30 30 39 01 01 :68
C870: 01 01 01 01 01 01 01 01 01 01 07 01 01 :16
C880: 01 A8 A9 AA AB 01 01 01 01 01 07 01 01 :88
C890: 01 01 01 01 01 01 01 01 01 01 07 01 01 :16
C8A0: 30 30 30 30 30 30 30 30 39 01 01 07 01 :97
C8B0: 01 01 01 01 01 01 01 01 07 01 01 07 01 :1C
C8C0: 61 16 01 01 01 01 01 07 01 01 07 01 01 :91
C8D0: 03 0A 01 01 A2 01 01 07 A0 01 07 01 A2 :08
C8E0: 03 0A 01 01 01 01 01 07 01 01 07 01 01 :27
C8F0: 03 5C 01 01 01 01 01 07 01 01 07 01 01 :79
sum : C3 E5 85 87 B9 6E 6E 3A AF E4 7E 10 B1 5A 10 :CF

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C900: 03 5D 01 01 01 01 01 07 01 01 07 01 01 :7A
C910: 03 0A 01 01 01 01 01 07 01 01 07 01 01 :27
C920: 03 0A 01 01 01 01 01 07 01 01 07 01 01 :27
C930: 03 0A 01 01 01 01 01 07 01 01 37 30 30 :42
C940: FF 84 01 01 01 01 01 07 01 01 01 01 01 :97
C950: 03 88 80 01 01 01 01 07 01 01 3A 30 30 :42
C960: 03 03 88 80 01 01 01 07 01 01 07 01 01 :26
C970: 6F 03 03 88 86 01 01 07 01 01 07 01 01 :9A
C980: 03 6F 03 03 0A 01 01 07 01 01 07 01 98 :30
C990: 03 03 6F 03 0A 01 01 07 01 01 07 01 9A :32
C9A0: 03 03 03 03 54 01 01 07 01 01 07 01 98 :0E
C9B0: 03 03 03 03 55 01 01 07 01 01 07 01 01 :78
C9C0: 03 03 03 03 0A 01 01 07 01 01 07 01 01 :2D
C9D0: FE 03 03 03 0A 01 01 07 01 01 07 01 01 :28
C9E0: 03 FE 03 03 90 01 01 07 01 01 07 01 01 :AE
C9F0: 03 03 FD 03 0A 01 01 07 01 01 07 01 01 :27
sum : 93 0C 8E 26 F8 10 10 70 10 10 CD 6E 05 07 05 :B5

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CA00: 03 03 03 03 0A 01 01 07 01 01 37 30 30 :48
CA10: 03 03 03 03 0A 01 01 07 01 01 01 01 B4 :8E
CA20: 11 5A 58 11 18 01 01 37 30 39 01 3A 30 :8C
CA30: 01 01 01 03 01 01 01 01 01 07 01 07 01 :EE
CA40: 01 01 01 01 01 01 01 01 01 07 01 07 01 :1C
CA50: 01 6C 6A 6A 6C 01 01 01 01 07 01 07 01 :C4
CA60: 01 05 88 8A 05 01 01 01 01 07 01 07 01 :9A
CA70: 01 05 89 8B 05 01 01 01 01 07 01 07 01 :96
CA80: 01 6D 68 6B 6D 01 01 01 01 07 01 07 01 :C8
CA90: 01 01 01 01 01 01 01 01 01 07 01 07 01 :1C
CAA0: 01 01 01 01 01 01 01 01 01 07 01 07 01 :1C
CAB0: 01 01 01 01 01 01 01 01 01 07 01 07 01 :1C
CAC0: 30 30 30 30 30 30 30 30 30 38 01 07 01 :F4
CAD0: 01 01 01 01 01 01 01 01 A2 01 A2 07 01 :58
CAE0: 30 30 30 30 30 30 30 30 30 30 38 01 01 :4C
CAF0: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
sum : 82 AA 0E 9A 76 6E 6E B0 3E E4 16 F1 21 22 :1E

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CB00: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
CB10: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
CB20: 01 01 01 01 01 01 01 01 01 01 01 9E 01 :48
CB30: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
CB40: 01 01 4A 40 40 40 40 40 40 40 40 49 01 :17
CB50: 01 01 08 04 04 04 04 04 04 04 04 08 01 :3C
CB60: 01 01 08 04 04 04 68 68 68 68 04 04 01 :08
CB70: 01 01 08 04 04 04 01 01 01 01 04 04 01 :30
CB80: 01 01 08 04 04 04 01 01 01 01 04 04 01 :E0
CB90: 01 01 08 04 04 04 01 01 01 01 04 04 01 :E2
CBA0: 01 01 08 04 04 04 01 01 01 01 04 04 01 :30
CBB0: 01 01 08 04 04 04 6A 6A 6A 6A 04 04 01 :04
CBC0: 01 01 08 68 68 68 68 68 68 68 08 01 01 :42
CBD0: 40 40 45 40 40 40 40 49 4A 40 40 40 45 :10
CBE0: 01 01 01 01 01 01 01 08 08 01 01 01 01 :F6
CBF0: 6C 6C 6C 6C 6C 01 01 08 08 01 01 6C 6C :4C
sum : BA BA 40 78 78 0D CE 94 99 CE 0D 15 78 DA BA :57

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CC00: 05 05 05 05 05 01 01 08 08 01 01 05 05 :46
CC10: 05 05 05 05 05 01 01 08 08 01 01 05 05 :46
CC20: 05 05 05 05 05 01 01 08 08 01 01 05 05 :46
CC30: 05 05 05 05 05 01 01 08 08 01 01 05 05 :46
CC40: 05 05 05 05 05 01 01 08 08 01 E9 05 05 :2E
CC50: 05 05 05 05 05 01 01 08 08 E5 01 05 05 :FA
CC60: 6D 6D 6D 6D 6D 01 01 08 08 01 01 05 05 :FE
CC70: 01 01 01 01 01 01 01 08 08 01 01 05 05 :E4
CC80: 01 01 01 01 01 01 01 08 08 01 01 05 05 :32
CC90: 40 40 40 40 40 40 40 44 08 01 01 05 05 :27
CCA0: 01 01 01 01 01 01 3C 08 08 3C 01 05 05 :A8
CCB0: 01 01 01 01 01 01 01 08 08 01 01 05 05 :32
CCC0: 6C 6C 6C 6C 6C 01 01 08 08 01 01 05 05 :49
CCD0: 05 05 05 05 05 01 01 08 08 01 01 05 05 :4F
CCE0: 05 6D 6D 05 05 01 01 08 08 01 01 05 05 :E6
CCF0: 05 DC DE 05 05 01 01 08 08 01 01 05 05 :A6
sum : 4A 89 8B 4A 53 4F 8A BC 80 2F F8 50 50 A6 AC :79

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CD00: 05 DD DF 05 05 01 01 08 08 01 01 05 05 :AA
CD10: 05 05 05 05 05 01 01 08 08 01 01 05 05 :46
CD20: 05 05 05 05 05 01 01 08 08 01 F4 05 05 :39
CD30: 6D 6D 6D 6D 6D 01 01 08 08 01 01 6D 6D :56
CD40: 01 01 01 01 01 01 01 08 08 01 01 01 01 :1E
CD50: 01 01 01 01 01 01 01 08 08 E3 01 01 01 :00
CD60: 01 01 01 01 01 01 01 08 47 40 40 40 40 :16
CD70: 01 01 01 01 01 01 01 47 40 40 40 40 40 :4E
CD80: 01 01 01 01 01 01 01 01 01 01 01 01 01 :10
CD90: 30 30 30 30 30 39 01 01 01 01 3A 30 30 :57
CDA0: CC CA CE D6 D7 07 01 01 01 01 07 D6 D7 :34
CDB0: 01 F5 01 01 01 07 01 01 01 01 07 01 01 :10
CDC0: F8 01 01 01 01 07 01 01 01 01 07 01 01 :13
CDD0: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CDE0: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CDF0: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
sum : 79 4C 5E 8C 8D 6C 10 87 BF 70 DE 0A 0B D8 D8 :13

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CE00: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CE10: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CE20: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CE30: 01 01 01 01 01 07 01 01 01 01 07 01 01 :1C
CE40: 30 39 01 01 01 33 30 30 30 30 34 01 01 :00
CE50: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CE60: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CE70: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CE80: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CE90: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CEA0: 04 07 01 01 01 07 3C F8 F8 3C 07 01 01 :92
CEB0: 04 07 01 01 01 33 30 30 30 30 34 01 01 :43
CEC0: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
CED0: 04 07 01 A8 01 07 01 01 01 01 07 01 04 :7C
CEE0: 04 07 AA 01 01 07 01 01 01 01 07 01 04 :80
CEF0: 04 07 01 01 01 07 01 01 01 01 07 01 04 :2E
sum : 60 8A B9 B7 10 C8 A9 65 65 A9 CA 10 B7 B9 8B :83

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
CF00: 68 32 01 01 01 32 01 01 01 01 32 01 68 :A8
CF10: 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D :00
CF20: 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CF30: 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
CF40: 00 26 2C 2C 2C 2C 2C 2C 2C 2C 2C 27 00 :50
CF50: 00 0D 17 10 16 08 01 01 01 01 08 17 10 :A7
CF60: 00 0D 08 03 0A 08 4C 6C 6C 4C 08 0B 03 :C9
CF70: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CF80: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CF90: 00 0D 08 03 54 08 01 05 05 01 08 56 03 :FA
CFA0: 00 0D 08 03 55 08 01 05 05 01 08 57 03 :FC
CFB0: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CFC0: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CFD0: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CFE0: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
CFF0: 00 0D 08 03 0A 08 01 05 05 01 08 0B 03 :65
sum : 98 14 DF 88 69 E3 B0 F4 F4 B0 E3 76 8B D4 0A :FE

```

## リスト7-11 GAMMA5 IPLプログラム "IPL"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: 21 40 C0 11 00 8F 01 50 00 ED B0 00 21 00 01 :D2
C010: 02 00 16 7F AF 3C E5 D5 C5 CD 9A 36 C1 D1 E1 :49
C020: F3 15 28 0D 0C 24 79 FE 11 20 E9 04 04 0E 01 :2D
C030: E3 C3 00 8F 00 00 00 00 00 00 00 00 00 00 :35
C040: 21 00 90 01 01 12 16 50 AF 3C E5 D5 C5 CD 9A :32
C050: C1 D1 E1 38 F3 15 28 0D 0C 24 79 FE 11 20 E9 :AD
C060: 04 0E 01 18 E3 00 00 00 00 00 00 00 00 00 :0E
C070: F3 3A C2 E6 F6 02 D3 31 AF D3 51 D8 32 E6 DF :49
C080: 32 C3 00 01 00 00 00 00 00 00 00 00 00 00 :F6
C090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
C0F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 04 F4 32 64 88 18 70 B1 40 0D E2 E8 EE B2 45 :A9

```





## リスト7-12 GAMMA 5 マップエディタ用マシン語メインプログラム "EDIT"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C000: C3 67 3C C3 8D 3B AF C5 E5 D3 5C 77 23 10 FC E1 :00
C010: C1 C5 E5 D3 5D 77 23 10 FC E1 C1 C5 E5 D3 5E 77 :35
C020: 23 10 FC E1 C1 CD 72 39 0D 20 DC D3 5F C9 CD 55 :6F
C030: 38 D8 CD 3D 38 CD 72 39 0D 20 DC D3 5F C9 CD 55 :E7
C040: D3 5C 77 23 10 FC E1 C1 C5 E5 D3 5D 77 23 10 FC :F7
C050: E1 C1 D3 5F C9 E5 D5 11 00 C0 A7 ED 52 D1 E1 C9 :89
C060: C5 01 03 00 C5 D5 D3 5C ED 80 D1 C1 C5 D5 D3 5D :88
C070: ED 80 D1 C1 D5 D3 5E ED 80 D3 5F D1 C1 C9 C5 01 :25
C080: 03 00 C5 D5 D3 5C ED 80 D1 C1 D5 D3 5D ED 80 D3 :70
C090: 5F D1 C1 C9 C5 D5 01 03 00 D3 5E ED 80 D3 5F D1 :29
C0A0: C1 C9 CD B2 38 00 06 0C CD 7E 38 CD 6A 39 D8 10 :FE
C0B0: F7 C9 E5 D5 21 00 C0 A7 ED 52 D1 E1 C9 3A 07 E0 :0D
C0C0: A7 28 0F FE 10 30 28 06 0C CD 60 38 CD 6A 39 10 :3B
C0D0: F8 C9 06 0C E5 21 00 C0 A7 ED 52 E1 30 05 CD 60 :C2
C0E0: 38 18 06 C5 01 09 00 09 C1 CD 6A 39 10 E6 C9 CD :EB
C0F0: B2 38 00 06 0C 18 14 06 0C E5 21 7E FE A7 E0 52 :72
sum : E8 86 2B F1 49 48 8D 9D 33 09 3C 1F CA 32 3F A2 :89

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C100: E1 D8 E5 21 00 C0 A7 ED 52 E1 D0 CD 60 38 CD 6A :B2
C110: 39 D8 10 F7 C9 3A 07 E0 A7 28 0F FE 10 30 25 06 :49
C120: 0C CD 94 38 CD 6A 39 10 F8 C9 06 0C E5 21 00 C0 :BE
C130: A7 ED 52 E1 30 05 CD 94 38 18 03 23 23 23 CD 6A :50
C140: 39 10 E9 C9 CD B2 38 D0 06 0C 18 14 06 0C E5 21 :D8
C150: 7E FE A7 ED 52 E1 D8 E5 21 00 C0 A7 ED 52 E1 D0 :78
C160: CD 94 38 CD 6A 39 D8 10 F7 C9 E5 21 50 00 19 E8 :0B
C170: E1 C9 D5 11 50 00 19 D1 C9 D5 11 A0 00 19 D1 C9 :CC
C180: D5 5D 16 C0 2E 50 CD C2 39 19 D1 C9 CB 24 CB 24 :DF
C190: 2C 2C 2C 18 EB C5 D5 21 43 BC 11 50 00 3A 02 E0 :BE
C1A0: A7 28 04 47 19 10 FD 11 C0 03 3A 09 E0 A7 28 04 :0A
C1B0: 47 19 10 FD EB 3A 08 E0 6F 26 03 CD C2 39 19 D1 :C4
C1C0: C1 C9 F5 C5 AF 06 08 CB 1D 30 01 84 1F 10 F8 CB :90
C1D0: 1D 67 C1 F1 C9 C5 D5 06 07 11 00 40 0E C0 7D 91 :D3
C1E0: 38 04 6F 7A 83 5F CB 0A CB 09 10 F2 65 6B D1 C1 :14
C1F0: C9 3A 02 E0 CB 3F CB 3F 47 7C 90 30 01 AF 67 E5 :7B
sum : 00 0D F5 F1 82 FD CF F5 F1 58 76 4B BB 4B 2A 1A :8A

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C200: CD D5 39 4D 7D 32 08 E0 7C 32 0A E0 06 00 2A 00 :87
C210: E0 09 EB E1 6C CD D5 39 2C 7D 32 09 E0 07 07 07 :D5
C220: 17 30 02 06 01 4F EB 09 7E C9 D5 7C FE 90 30 0B :F4
C230: 67 2E 24 CD C2 39 11 00 41 18 1D FE F0 30 0D 06 :09
C240: 90 67 2E 7D CD C2 39 11 00 56 18 0C FE FA 30 04 :14
C250: 26 01 18 D0 26 03 18 D9 19 D1 C9 21 03 C0 22 04 :F3
C260: E0 21 00 01 22 06 E0 ED 5B 04 E0 2A 00 E0 3A 06 :80
C270: E0 4F 06 00 09 3A 07 E0 07 07 07 17 30 01 04 4F :0F
C280: 09 7E 67 F5 CD 2A 3A F1 FE 90 30 05 CD 15 39 18 :FB

```

```

C290: 18 FE F0 38 0F FE FA 30 04 3E 01 18 02 3E 03 CD :E0
C2A0: 15 39 18 05 F5 CD BD 38 F1 2A 04 E0 23 23 23 22 :AC
C2B0: 04 E0 3A 06 E0 3C 32 06 E0 D6 10 20 AA 32 06 E0 :20
C2C0: 3A 07 E0 3C 32 07 E0 2A 04 E0 01 90 03 09 22 04 :47
C2D0: E0 3A 07 E0 D6 13 20 8F C9 00 AF 32 09 3A 3A 02 :92
C2E0: E0 A7 20 0E 21 03 C0 22 04 E0 21 00 01 22 06 E0 :C9
C2F0: 18 13 6F 26 50 CD C2 39 11 43 BC 19 22 04 E0 21 :28
sum : ED A4 B5 D7 F4 A7 B6 4C 97 93 C8 C9 A0 73 A5 33 :60

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C300: 00 00 22 06 E0 ED 5B 04 E0 2A 00 E0 3A 06 E0 4F :AD
C310: 06 00 09 3A 07 E0 07 07 07 17 30 01 04 4F 09 7E :67
C320: FE 10 38 27 FE F0 30 23 67 F5 CD 2A 3A F1 FE 90 :BA
C330: 30 0C CD 15 39 3A D9 3A 3C 32 D9 3A 18 0D F5 CD :0C
C340: BD 38 F1 3A D9 3A 3C 3C 32 D9 3A 2A 04 E0 23 23 :44
C350: 23 22 04 E0 3A 06 E0 3C 32 06 E0 D6 10 20 A6 32 :7B
C360: 06 E0 3A 07 E0 3C 32 07 E0 2A 04 E0 01 90 03 09 :07
C370: 22 04 E0 3A 07 E0 D6 13 20 8B 3A D9 3A FE 46 D0 :1C
C380: 47 3E 46 90 C8 06 F0 10 FE 3D 20 F9 C9 CD 5C 3C :AB
C390: 2A FE BF 22 00 E0 21 03 C0 22 04 E0 21 00 01 22 :17
C3A0: 06 E0 ED 5B 04 E0 2A 00 E0 3A 06 E0 4F 06 00 09 :9A
C3B0: 3A 07 E0 07 07 07 17 30 01 04 4F 09 7E 67 F5 CD :81
C3C0: 2A 3A F1 FE 90 30 05 CD 15 39 18 0D FE F0 38 04 :82
C3D0: D6 80 18 E9 F5 CD BD 38 F1 2A 04 E0 23 23 23 22 :98
C3E0: 04 E0 3A 06 E0 3C 32 06 E0 D6 10 20 B5 32 06 E0 :2B
C3F0: 3A 07 E0 3C 32 07 E0 2A 04 E0 01 90 03 09 22 04 :47
sum : 2B 1E 34 14 82 60 B5 72 77 B2 D4 5D 6F 69 C3 96 :25

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
C400: E0 3A 07 E0 D6 13 20 9A C9 2A 00 E0 3A 02 E0 A7 :3A
C410: 28 09 FE 04 28 05 FE 08 28 01 C9 01 10 00 09 06 :78
C420: 00 7E FE 90 38 32 FE B0 30 0D 4F E6 03 FE 03 20 :8A
C430: 25 79 E6 FC 4F 18 20 FE C8 38 1D FE D0 30 19 4F :88
C440: FE CA 38 0A E6 01 20 06 E0 5F E6 0F 20 0A 79 E6 :E1
C450: 01 28 03 0D 18 01 0C 71 23 10 C6 C9 21 03 C0 06 :7B
C460: 30 0E C8 CD 06 38 C9 21 00 E0 36 00 11 01 E0 01 :04
C470: 00 02 ED B0 CD 5C 3C 21 E0 CE 22 00 E0 AF 32 02 :88
C480: E0 CD 5B 3A 3A 02 E0 3C 3C 32 02 E0 D6 0C 20 18 :04
C490: 32 02 E0 2A 00 E0 11 10 00 A7 ED 52 22 00 E0 11 :38
C4A0: 00 C0 A7 ED 52 30 01 C9 CD 09 3C CD DA 3A 18 04 :7F
C4B0: 25 79 E6 FC 4F 18 20 FE C8 38 1D FE D0 30 19 4F :88
C4C0: FE CA 38 0A E6 01 20 06 E0 5F E6 0F 20 0A 79 E6 :E1
C4D0: 01 28 03 0D 18 01 0C 71 23 10 C6 C9 21 03 C0 06 :7B
C4E0: 30 0E C8 CD 06 38 C9 21 00 E0 36 00 11 01 E0 01 :04
C4F0: 00 02 ED B0 CD 5C 3C 21 E0 CE 22 00 E0 AF 32 02 :88
sum : C2 46 91 E5 02 B8 B0 D5 9A C1 85 72 23 20 CC 46 :67

```

## リスト7-13 GAMMA 5 マップエディタ用マシン語プログラム① "GRGETT"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
BE00: 00 2A F0 BE ED 5B F4 BE ED 4B F2 BE D5 C5 7E D3 :A5
BE10: 5E 12 D3 5F 23 13 10 F6 C1 D1 E5 21 50 00 19 EB :CA
BE20: E1 00 20 E8 00 C9 00 00 00 00 00 00 00 00 00 :BF
BE30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BE40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BE50: 00 2A F0 BE ED 5B F4 BE ED 4B F2 BE D5 C5 7E D3 :A5
BE60: 5C 12 D3 5F 23 13 10 F6 C1 D1 E5 21 50 00 19 EB :CA
BE70: D3 5F 23 13 10 F6 C1 D1 E5 21 50 00 19 EB E1 0D :48
BE80: 20 DA 00 C9 00 00 00 00 00 00 00 00 00 00 00 :C3
BE90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BEA0: 00 2A F0 BE ED 5B F4 BE ED 4B F2 BE D5 C5 7E D3 :A5
BEB0: 5C 12 D3 5F 23 13 10 F6 C1 D1 E5 21 50 00 19 EB :CA
BEC0: D3 5F 23 13 10 F6 C1 D1 E5 21 50 00 19 EB E1 0D :48
BED0: 23 13 10 F6 C1 D1 E5 21 50 00 19 EB E1 0D 20 CC :02
BEE0: 00 C9 00 00 00 00 00 00 00 00 00 00 00 00 :C9
BEF0: 00 C0 0C 03 57 C3 00 00 00 00 00 00 00 00 :E9
sum : E0 F5 CB 27 68 93 73 DF 14 3A 4C A3 23 FF 21 C0 :54

```

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
BF00: F3 3A C2 E6 F5 F6 02 D3 31 CD 00 BE F1 D3 31 FB :41
BF10: C9 00 00 00 00 00 00 00 00 00 00 00 00 00 :C9
BF20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF50: F3 3A C2 E6 F5 F6 02 D3 31 CD 50 BE F1 D3 31 FB :91
BF60: C9 00 00 00 00 00 00 00 00 00 00 00 00 00 :C9
BF70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BF90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BFA0: F3 3A C2 E6 F5 F6 02 D3 31 CD A0 BE F1 D3 31 FB :E1
BFB0: C9 00 00 00 00 00 00 00 00 00 00 00 00 00 :C9
BFC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BFD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BFE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BFF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 34 AE 46 B2 DF E2 06 79 93 67 F0 3A D3 79 93 F1 :0E

```

## リスト7-14 GAMMA 5 マップエディタ用マシン語プログラム② "ZSCENT"

```

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :sum
D100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D150: F3 3A C2 E6 F5 F6 02 D3 31 00 00 00 CD 03 38 F1 :BF
D160: D3 31 FB C9 00 00 00 00 00 00 00 00 00 00 :C8
D170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D1A0: F3 3A C2 E6 F5 F6 02 D3 31 AF D3 51 CD 00 38 F1 :8F
D1B0: D3 31 FB C9 00 00 00 00 00 00 00 00 00 00 :C8
D1C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D1D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D1E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
D1F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
sum : 8C D6 7A 5E EA EC 04 A6 62 AF D3 51 9A 03 70 E2 :DE

```





## リスト7-15 GAMMA 5 マップエディタ用BASICプログラム

```

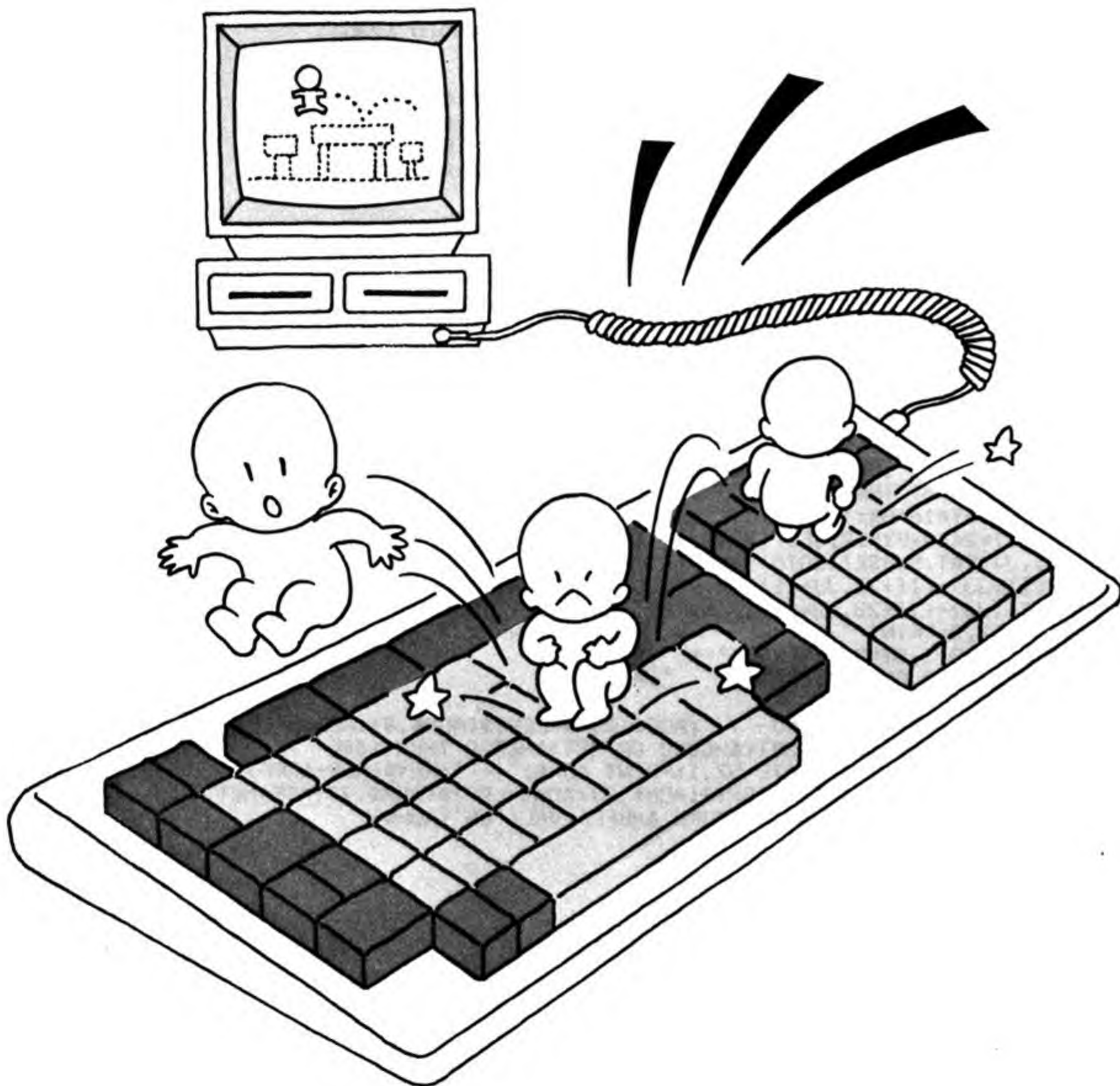
1000 'ZENON MAP CONSTACTION 85'11/06 KURE EIJI
1010 WIDTH 80,25:CONSOLE 0,25,0,1
1020 CLEAR ,&HBDFF:BLOAD 'GRGETT':BLOAD 'ZSCENT':DEFINT A-Z:DIM C(255),PAT.S(10
0)
1040 FOR J=40 TO 179 STEP 14:FOR I=440 TO 560 STEP 26
1050 VIEW(I,J)-(I+23,J+11),,1:NEXT:NEXT:VIEW(0,0)-(639,199)
1060 GOSUB *KAKU
1090 GRGET1=&HBF00:GRGET2=&HBF50:GRGET3=&HBFA0:ZSC1=&HD1A0:ZSC2=&HD150
1100 MTOP=&HC000:CTOP=&H4100:CTOP1=&H5600:CX=0:CY=0:CNO=0
1120 FOR I=0 TO &H8F:C(I)=CTOP+I*36:NEXT
1130 FOR I=&H90 TO &HEF:C(I)=CTOP1+(I-&H90)*&H70:NEXT
1140 FOR I=&HF0 TO &HFF:C(I)=CTOP+(I-&H80)*36:NEXT
1150 LOCATE 57,1:PRINT 'C000 - COFF':POKE &HBFFE,&HF0:POKE &HBFFF,&HBF:CALL ZSC2
1160 N=0:LOCATE 57,2:PRINT 'CHR ';N;'-';N+49:CSE=0:MSE=&HC000:GOSUB 1710
1170 LOCATE 56,0:PRINT 'COMMAND M,G,S,T ';
1180 K$=INPUT$(1)
1190 K=INSTR('MmGgSsTt',K$)/2:ON K GOSUB *GETMAP,*GETCHR,*SETCHR,*SCTEST
1200 GOTO 1170
1210 *SETCHR:GX=8:GY=8
1215 JJ=CY*14+40:II=CX*26+440
1220 LOCATE 57,3:PRINT 'SELECT NO. '
1230 K$=INKEY$:IF K$="" THEN GOSUB 1470:GOSUB 1460:GOTO 1230
1240 IF K$=CHR$(13) THEN 1480
1250 IF K$="5" THEN 1340
1260 IF K$="4" THEN IF CX>0 THEN GOSUB 1470:GX=GX-1:GOTO 1310
1270 IF K$="6" THEN IF CX<4 THEN GOSUB 1470:GX=GX+1:GOTO 1310
1280 IF K$="2" THEN IF CY<9 THEN GOSUB 1470:CY=CY+1:GOTO 1310
1290 IF K$="8" THEN IF CY>0 THEN GOSUB 1470:CY=CY-1:GOTO 1310
1300 GOTO 1230
1310 CNO=GX*10+CY
1320 JJ=CY*14+40:II=CX*26+440
1330 GOSUB 1460:GOTO 1230
1340 LOCATE 57,3:PRINT 'MOVE & SET ':GOSUB 1490
1350 K$=INKEY$:IF K$="" THEN FOR Q=0 TO 10:NEXT:GOSUB 1500:FOR Q=0 TO 10:NEXT:G
OSUB 1490:GOTO 1350
1360 IF K$="" THEN GOSUB 1500:GOSUB 1480:GOTO 1220
1370 IF K$="5" THEN GOSUB 1500:GOTO 1430
1380 IF K$="4" THEN IF GX>0 THEN GOSUB 1500:GX=GX-1:GOSUB 1490:GOTO 1350
1390 IF K$="6" THEN IF GX<15 THEN GOSUB 1500:GX=GX+1:GOSUB 1490:GOTO 1350
1400 IF K$="2" THEN IF GY<15 THEN GOSUB 1500:GY=GY+1:GOSUB 1490:GOTO 1350
1410 IF K$="8" THEN IF GY>0 THEN GOSUB 1500:GY=GY-1:GOSUB 1490:GOTO 1350
1420 IF K$=CHR$(13) THEN GOSUB 1500:GOTO 1480 ELSE GOTO 1350
1430 MADR=MSE+GY*16+GX:IF CNO+CSE<&H100 THEN POKE MADR,CNO+CSE
1440 I=(GX+1)*24:J=GY*12:GET(II,JJ)-(STEP(23,11),PAT.S
1450 PUT(I,J),PAT.S,PSET:GOTO 1350
1460 VIEW(II,JJ)-(II+23,JJ+11),,7:VIEW(0,0)-(639,199):RETURN
1470 VIEW(II,JJ)-(II+23,JJ+11),,1:VIEW(0,0)-(639,199):RETURN
1480 LOCATE 57,3:PRINT '':RETURN
1490 LOCATE GX*3+3,GY*3/2:PRINT ' * ':RETURN
1500 LOCATE GX*3+3,GY*3/2:PRINT '':RETURN
1510 *GETMAP
1520 LOCATE 57,3:INPUT '(C000- ) ':ADR$:ADH$=LEFT$(ADR$,3):ADR$=ADH$+'0'
1530 MET=VAL('&H'+ADR$):IF MET<&HC000 OR MET>&HCF20 THEN 1480
1540 ADR=VAL('&H'+ADR$):LOCATE 57,1:PRINT ADR$;' - ':HEX$(ADR+&HFF)
1550 ADM$=HEX$(ADR-16):ADL$=RIGHT$(ADM$,2):ADH$=LEFT$(ADM$,2):MSE=MET
1560 POKE &HBFFE,VAL('&H'+ADL$):POKE &HBFFF,VAL('&H'+ADH$)
1570 CALL ZSC2
1580 GOTO 1480
1600 *SCTEST
1610 LOCATE 57,3:PRINT 'TEST OK ? (Y/N)'
1620 K$=INKEY$:IF K$="Y" OR K$="y" THEN 1650
1630 IF K$="N" OR K$="n" THEN 1480 ELSE 1620
1650 CALL ZSC1:WIDTH 80
1660 GOSUB *KAKU:GOTO 1150
1670 *GETCHR
1680 LOCATE 57,3:INPUT 'TOP CHR NO.':N=N+INT(N)
1690 IF N<0 OR N>211 THEN 1480
1700 LOCATE 57,2:PRINT 'CHR ';N;'-';N+49;' ':CSE=N
1710 FOR I=0 TO 49:IF N>&HFF THEN 1820
1720 MET=C(N):MET$=HEX$(MET)
1730 MEL$=RIGHT$(MET$,2):MEH$=LEFT$(MET$,2)
1740 POKE &HBEF0,VAL('&H'+MEL$):POKE &HBEF1,VAL('&H'+MEH$)
1750 III=INT(I/10):JJJ=I-III*10

```



```

1760 JJ=JJJ*14+40:II=III*26+440
1770 POKE &HBEF4,&H57:POKE &HBEF5,&HC3
1780 IF N<&H90 THEN CALL GRGET1:GOTO 1800
1790 IF N<&HF0 THEN CALL GRGET3:GOTO 1800 ELSE CALL GRGET1
1795 IF N=&HF0 THEN LINE(440,10)-STEP(24,12),0,BF
1796 CALL GRGET1
1800 GET(440,10)-STEP(23,11),PAT.S:PUT(II,JJ),PAT.S,PSET
1810 N=N+1
1820 NEXT
1830 LINE(440,10)-STEP(24,12),0,BF
1840 GOTO 1480
1900 *KAKU
1930 J=0:FOR I=0 TO 22 STEP 3:LOCATE 0,I:PRINT J:LOCATE 0,I+2:PRINT J+1:J=J+2:N
EXT
1960 LOCATE 57,4:PRINT '0 1 2 3 4'
1970 LOCATE 72,5:PRINT '0'
1980 LOCATE 72,14:PRINT '5'
1990 RETURN
    
```



アクションゲーム用キーボード



- A  $\mu$ COM-82インストラクション・セット
- B  $\mu$ COM-82マシン語 $\longleftrightarrow$ ニーモニック対応表
- C  $\mu$ COM-82ニーモニック $\longleftrightarrow$ マシン語対照表
- D 10進 $\longleftrightarrow$ 16進変換表
- E 1バイト符号付16進数

# 付録



# 付録A $\mu$ COMインストラクション・セット

命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	OPコード		
			S	Z	H	P/V	N	C			76	543	210
8 ビット・ ロード 命令	LD r, r'	$r \leftarrow r'$	.	.	.	.	.	.	1	4	01	r	r' ㊦
	LD r, n	$r \leftarrow n$	.	.	.	.	.	.	2	7	00	r	110 ㊦
											$\leftarrow$	n	$\rightarrow$
	LD r, (HL)	$r \leftarrow (HL)$	.	.	.	.	.	.	1	7	01	r	110 ㊦
	LD r, (IX+d)	$r \leftarrow (IX+d)$	.	.	.	.	.	.	3	19	11	011	101
											01	r	110 ㊦
											$\leftarrow$	d	$\rightarrow$
	LD r, (IY+d)	$r \leftarrow (IY+d)$	.	.	.	.	.	.	3	19	11	111	101
											01	r	110 ㊦
											$\leftarrow$	d	$\rightarrow$
	LD (HL), r	$(HL) \leftarrow r$	.	.	.	.	.	.	1	7	01	110	r ㊦
	LD (IX+d), r	$(IX+d) \leftarrow r$	.	.	.	.	.	.	3	19	11	011	101
											01	110	r ㊦
											$\leftarrow$	d	$\rightarrow$
	LD (IX+d), r	$(IY+d) \leftarrow r$	.	.	.	.	.	.	3	19	11	111	101
											01	110	r ㊦
											$\leftarrow$	d	$\rightarrow$
	LD (HL), n	$(HL) \leftarrow n$	.	.	.	.	.	.	2	10	00	110	110
											$\leftarrow$	n	$\rightarrow$
	LD (IX+d), n	$(IX+d) \leftarrow n$	.	.	.	.	.	.	4	19	11	011	101
											00	110	110
											$\leftarrow$	d	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
16 ビット・ ロード命令	LD (IY+d), n	$(IY+d) \leftarrow n$	.	.	.	.	.	.	4	19	11	111	101
											00	110	110
											$\leftarrow$	d	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD A, (BC)	$A \leftarrow (BC)$	.	.	.	.	.	.	1	7	00	001	010
	LD A, (DE)	$A \leftarrow (DE)$	.	.	.	.	.	.	1	7	00	011	010
	LD A, (nn)	$A \leftarrow (nn)$	.	.	.	.	.	.	3	13	00	111	010
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD (BC), A	$(BC) \leftarrow A$	.	.	.	.	.	.	1	7	00	000	010
	LD (DE), A	$(DE) \leftarrow A$	.	.	.	.	.	.	1	7	00	010	010
	LD (nn), A	$(nn) \leftarrow A$	.	.	.	.	.	.	3	13	00	110	010
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD A, I	$A \leftarrow I$	1	1	0	IFF	0	.	2	9	11	101	101
											01	010	111
	LD A, R	$A \leftarrow R$	1	1	0	IFF	0	.	2	9	11	101	101
											01	011	111
	LD I, A	$I \leftarrow A$	.	.	.	.	.	.	2	9	11	101	101
											01	000	111
	LD R, A	$R \leftarrow A$	.	.	.	.	.	.	2	9	11	101	101
											01	001	111
16 ビット・ ロード命令	LD dd, nn	$dd \leftarrow nn$	.	.	.	.	.	.	3	10	00	dd0	001 ㊦
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD IX, nn	$IX, nn$	.	.	.	.	.	.	4	14	11	011	101
											00	100	001
											$\leftarrow$	n	$\rightarrow$
	LD IY, nn	$IY \leftarrow nn$	.	.	.	.	.	.	4	14	11	111	101
											00	100	001
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD HL, (nn)	$H \leftarrow (nn+1)$ $L \leftarrow (nn)$	.	.	.	.	.	.	3	16	00	101	010
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$
	LD dd, (nn)	$dd_H \leftarrow (nn+1)$ $dd_L \leftarrow (nn)$	.	.	.	.	.	.	4	20	11	101	101
											01	dd1	011 ㊦
											$\leftarrow$	n	$\rightarrow$
											$\leftarrow$	n	$\rightarrow$



命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	OPコード		
			S	Z	H	P/V	N	C			76	543	210
16 ビット・ ロード 命令	LD IX, (nn)	$IX_H \leftarrow (nn+1)$ $IX_L \leftarrow (nn)$	.	.	.	.	.	.	4	20	11 011 101 00 101 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	.	.	.	.	.	.	4	20	11 111 101 00 101 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	.	.	.	.	.	.	3	16	00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	.	.	.	.	.	.	4	20	11 101 101 01 dd0 011 <sup>Ⓐ</sup> $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	.	.	.	.	.	.	4	20	11 011 101 00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	.	.	.	.	.	.	4	20	11 111 101 00 100 010 $\leftarrow n \rightarrow$ $\leftarrow n \rightarrow$		
	LD SP, HL	$SP \leftarrow HL$	.	.	.	.	.	.	1	6	11 111 001		
	LD SP, IX	$SP \leftarrow IX$	.	.	.	.	.	.	2	10	11 011 101 11 111 001		
	LD SP, IY	$SP \leftarrow IY$	.	.	.	.	.	.	2	10	11 111 101 11 111 001		
	PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$	.	.	.	.	.	.	1	11	11 qq0 101 <sup>Ⓑ</sup>		
	PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$	.	.	.	.	.	.	2	15	11 011 101 11 100 101		
	PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$	.	.	.	.	.	.	2	15	11 111 101 11 100 101		
	POP qq	$qq_H \leftarrow (SP+1)$ $qq_L \leftarrow (SP)$	.	.	.	.	.	.	1	10	11 qq0 001 <sup>Ⓑ</sup>		
	POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$	.	.	.	.	.	.	2	14	11 011 101 11 100 001		
	POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$	.	.	.	.	.	.	2	14	11 111 101 11 100 001		
エク スチ エン ジ 命 令	EX DE, HL	$DE \leftrightarrow HL$	.	.	.	.	.	.	1	4	11 101 011		
	EX AF, AF'	$AF \leftrightarrow AF'$	.	.	.	.	.	.	1	4	00 001 000		
	EXX	$(BC \leftrightarrow BC')$ $(DE \leftrightarrow DE')$ $(HL \leftrightarrow HL')$	.	.	.	.	.	.	1	4	11 011 001		
	EX (SP), HL	$H \leftrightarrow (SP+1)$ $L \leftrightarrow (SP)$	.	.	.	.	.	.	1	19	11 100 011		
	EX (SP), IX	$IX_H \leftrightarrow (SP+1)$ $IX_L \leftrightarrow (SP)$	.	.	.	.	.	.	2	23	11 011 101 11 100 011		
ブ ロ ック 転 送 命 令	LDI	$(DE) \leftarrow (HL)$ $DE \leftarrow DE+1$ $HL \leftarrow HL+1$ $BC \leftarrow BC-1$	.	.	0	<sup>①</sup> ↑	0	.	2	16	11 101 101 10 100 000		
	LDIR	$(DE) \leftarrow (HL)$ $DE \leftarrow DE+1$ $HL$ $BC \leftarrow BC-1$ until $BC=0$	.	.	0	0	0	.	2	21 if $BC \neq 0$ 16 if $BC = 0$	11 101 101 10 110 000		

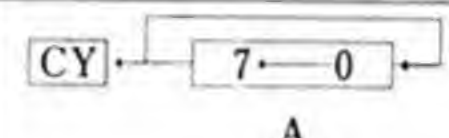
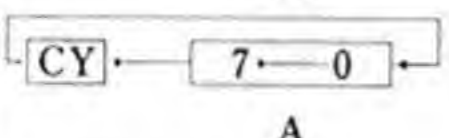
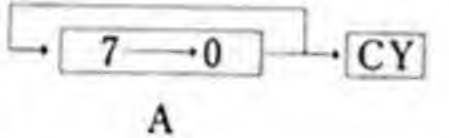
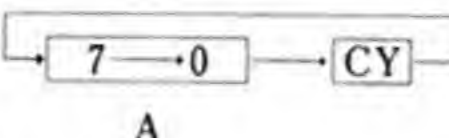


命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	OPコード		
			S	Z	H	P/V	N	C			76	543	210
ブロック転送命令	LDD	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1	·	·	0	1	0	·	2	16	11	101	101
	LDDR	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 until BC = 0	·	·	0	0	0	·	2	21 if BC ≠ 0 16 if BC = 0	11	101	101
ブロック・サーチ命令	CPI	A ← (HL) HL ← HL + 1 BC ← BC - 1	1	1	1	1	1	·	2	16	11	101	101
	CPIR	A ← (HL) HL ← HL + 1 BC ← BC - 1 until A = (HL) or BC = 0	1	1	1	1	1	·	2	21 if BC ≠ 0 and A ≠ (HL) 16 if BC = 0 or A = (HL)	11	101	101
	CPD	A ← (HL) HL ← HL - 1 BC ← BC - 1	1	1	1	1	1	·	2	16	11	101	101
	CPDR	A ← (HL) HL ← HL - 1 BC ← BC - 1 until A = (HL) or BC = 0	1	1	1	1	1	·	2	21 if BC ≠ 0 and A ≠ (HL) 16 if BC = 0 or A = (HL)	11	101	101
8ビット算術論理演算命令	ADD A, r	A ← A + r	1	1	1	V	0	1	1	4	10	000	r
	ADD A, n	A ← A + n	1	1	1	V	0	1	2	7	11	000	110
	ADD A, (HL)	A ← A + (HL)	1	1	1	V	0	1	1	7	10	000	110
	ADD A, (IX + d)	A ← A + (IX + d)	1	1	1	V	0	1	3	19	11	011	101
	ADD, A, (IY + d)	A ← A + (IY + d)	1	1	1	V	0	1	3	19	10	000	110
	ADC A, r	A ← A + r + CY	1	1	1	V	0	1	1	4	11	011	101
	ADC A, n	A ← A + n + CY	1	1	1	V	0	1	2	7	10	000	110
	ADC A, (HL)	A ← A + (HL) + CY	1	1	1	V	0	1	1	7	11	011	101
	ADC A, (IX + d)	A ← A + (IX + d) + CY	1	1	1	V	0	1	3	19	10	001	110
	ADC A, (IY + d)	A ← A + (IY + d) + CY	1	1	1	V	0	1	3	19	11	111	101
	SUB r	A ← A - r	1	1	1	V	1	1	1	4	10	010	r
	SUB n	A ← A - n	1	1	1	V	1	1	2	7	11	010	110
	SUB (HL)	A ← A - (HL)	1	1	1	V	1	1	1	7	10	010	110
	SUB (IX + d)	A ← A - (IX + d)	1	1	1	V	1	1	3	19	11	011	101
	SUB (IY + d)	A ← A - (IY + d)	1	1	1	V	1	1	3	19	10	010	110
	SBC A, r	A ← A - r - CY	1	1	1	V	1	1	1	4	11	111	101
	SBC A, n	A ← A - n - CY	1	1	1	V	1	1	2	7	10	010	110
	SBC A, (HL)	A ← A - (HL) - CY	1	1	1	V	1	1	1	7	11	011	110

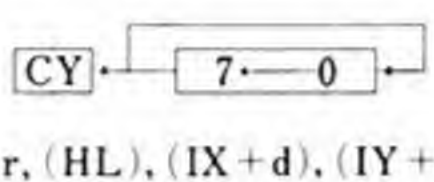
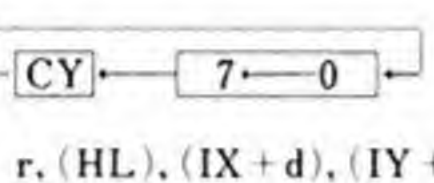
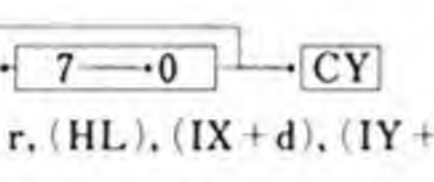
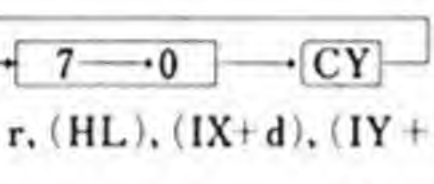
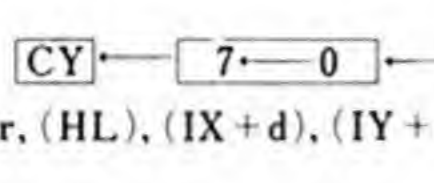


命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	OPコード		
			S	Z	H	P/V	N	C			76	543	210
8 ビ ッ ト 算 術 論 理 演 算 命 令	SBC A, (IX+d)	$A \leftarrow A - (IX+d) - CY$	1	1	1	V	1	1	3	19	11	011	101
											10	011	110
											←	d	→
	SBC A, (IY+d)	$A \leftarrow A - (IY+d) - CY$	1	1	1	V	1	1	3	19	11	111	101
											10	011	110
											←	d	→
	AND r	$A \leftarrow A \wedge r$	1	1	1	P	0	0	1	4	10	100	r ㊦
	AND n	$A \leftarrow A \wedge n$	1	1	1	P	0	0	2	7	11	100	110
											←	n	→
	AND (HL)	$A \leftarrow A \wedge (HL)$	1	1	1	P	0	0	1	7	10	100	110
	AND (IX+d)	$A \leftarrow A \wedge (IX+d)$	1	1	1	P	0	0	3	19	11	011	101
											10	100	110
											←	d	→
	AND (IY+d)	$A \leftarrow A \wedge (IY+d)$	1	1	1	P	0	0	3	19	11	111	101
											10	100	110
											←	d	→
	OR r	$A \leftarrow A \vee r$	1	1	0	P	0	0	1	4	10	110	rr ㊦
	OR n	$A \leftarrow A \vee n$	1	1	0	P	0	0	2	7	11	110	110
											←	n	→
	OR (HL)	$A \leftarrow A \vee (HL)$	1	1	0	P	0	0	1	7	10	110	110
	OR (IX+d)	$A \leftarrow A \vee (IX+d)$	1	1	0	P	0	0	3	19	11	011	101
											10	110	110
											←	d	→
	OR (IY+d)	$A \leftarrow A \vee (IY+d)$	1	1	0	P	0	0	3	19	11	111	101
											10	110	110
											←	d	→
	XOR r	$A \leftarrow A \oplus r$	1	1	0	P	0	0	1	4	10	101	r ㊦
	XOR n	$A \leftarrow A \oplus n$	1	1	0	P	0	0	2	7	11	101	110
											←	n	→
	XOR (HL)	$A \leftarrow A \oplus (HL)$	1	1	0	P	0	0	1	7	10	101	110
	XOR (IX+d)	$A \leftarrow A \oplus (IX+d)$	1	1	0	P	0	0	3	19	11	011	101
											10	101	110
											←	d	→
	XOR (IY+d)	$A \leftarrow A \oplus (IY+d)$	1	1	0	P	0	0	3	19	11	111	101
											10	101	110
											←	d	→
	CP r	$A \leftarrow r$	1	1	1	V	1	1	1	4	10	111	r ㊦
	CP n	$A \leftarrow n$	1	1	1	V	1	1	2	7	11	111	110
											←	n	→
	CP (HL)	$A \leftarrow (HL)$	1	1	1	V	1	1	1	7	10	111	110
	CP (IX+d)	$A \leftarrow (IX+d)$	1	1	1	V	1	1	3	19	11	011	101
											10	111	110
											←	d	→
	CP (IY+d)	$A \leftarrow (IY+d)$	1	1	1	V	1	1	3	19	11	111	101
											10	111	110
											←	d	→
	INC r	$r \leftarrow r + 1$	1	1	1	V	0	·	1	4	00	r	100 ㊦
	INC (HL)	$(HL) \leftarrow (HL) + 1$	1	1	1	V	0	·	1	11	00	110	100
	INC (IX+d)	$(IX+d) \leftarrow (IX+d) + 1$	1	1	1	V	0	·	3	23	11	011	101
											00	110	100
											←	d	→
	INC (IY+d)	$(IY+d) \leftarrow (IY+d) + 1$	1	1	1	V	0	·	3	23	11	111	101
											00	110	100
											←	d	→
	DEC r	$r \leftarrow r - 1$	1	1	1	V	1	·	1	4	00	r	101 ㊦
	DEC (HL)	$(HL) \leftarrow (HL) - 1$	1	1	1	V	1	·	1	11	00	110	101
	DEC (IX+d)	$(IX+d) \leftarrow (IX+d) - 1$	1	1	1	V	1	·	3	23	11	011	101
											00	110	101
											←	d	→

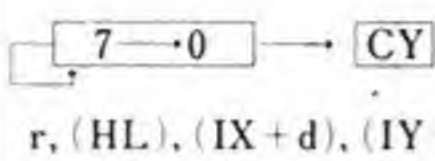
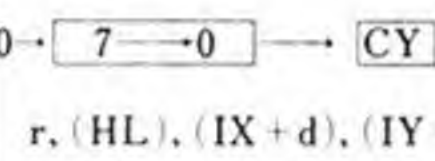
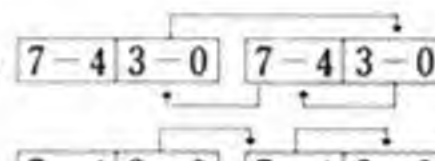
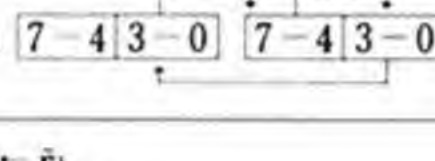


命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	O P コード		
			S	Z	H	P/V	N	C			76	543	210
演算命令 8ビット 演算命令	DEC (IY+d)	(IY+d)←(IY+d)−1	!	!	!	V	1	.	3	23	11 00 ←	111 110 d	101 101 →
16 ビ ッ ト 算 術 演 算 命 令	ADD HL, ss	HL←HL+ss	.	.	×	.	0	!	1	11	00	ss1	001 ㉠
	ADC HL, ss	HL←HL+ss+CY	!	!	×	V	0	!	2	15	11 01	101 ss1	101 010 ㉠
	SBC HL, ss	HL←HL−ss−CY	!	!	×	V	1	!	2	15	11 01	101 ss0	101 010 ㉠
	ADD IX, pp	IX←IX+pp	.	.	×	.	0	!	2	15	11 00	011 pp1	101 001 ㉡
	ADD IY, rr	IY←IY+rr	.	.	×	.	0	!	2	15	11 00	111 rr1	101 001 ㉢
	INC ss	ss←ss+1	.	.	.	.	.	.	1	6	00	ss0	011 ㉠
	INC IX	IX←IX+1	.	.	.	.	.	.	2	10	11 00	011 100	101 011
	INC IY	IY←IY+1	.	.	.	.	.	.	2	10	11 00	111 100	101 011
	DEC ss	ss←ss−1	.	.	.	.	.	.	1	6	00	ss1	011 ㉠
	DEC IX	IX←IX−1	.	.	.	.	.	.	2	10	11 00	011 101	101 011
	DEC IY	IY←IY−1	.	.	.	.	.	.	2	10	11 00	111 101	101 011
	ア キ ュ ム レ ー タ 操 作 命 令	DAA	Decimal adjust Acc	!	!	!	P	.	!	1	4	00	100
CPL		A← $\bar{A}$	.	.	1	.	1	.	1	4	00	101	111
NEG		A← $\bar{A}+1$	!	!	!	V	1	!	2	8	11 01	101 000	101 100
CCF		CY← $\bar{CY}$	.	.	×	.	0	!	1	4	00	111	111
SCF		CY←1	.	.	0	.	0	1	1	4	00	110	111
C P U コ ン ト ロ ー ル 命 令	NOP	No operation	.	.	.	.	.	.	1	4	00	000	000
	HALT	CPU halted	.	.	.	.	.	.	1	4	01	110	110
	DI	IFF←0	.	.	.	.	.	.	1	4	11	110	011
	EI	IFF←1	.	.	.	.	.	.	1	4	11	111	011
	IM 0	Set interrupt mode 0	.	.	.	.	.	.	2	8	11 01	101 000	101 110
	IM 1	Set interrupt mode 1	.	.	.	.	.	.	2	8	11 01	101 010	101 110
	IM 2	Set interrupt mode 2	.	.	.	.	.	.	2	8	11 01	101 011	101 110
ロ ー テ ー ト ・ シ フ ト 命 令	RLCA		.	.	0	.	0	!	1	4	00	000	111
	RLA		.	.	0	.	0	!	1	4	00	010	111
	RRCA		.	.	0	.	0	!	1	4	00	001	111
	RRA		.	.	0	.	0	!	1	4	00	011	111



命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	O P コード		
			S	Z	H	P/V	N	C			76	543	210
ロー ー テ ィ ト ・ シ フ ト 命 令	RLC r		1	1	0	P	0	1	2	8	11 001 011 00 000 r ㊦		
	RLC (HL)		1	1	0	P	0	1	2	15	11 001 011 00 000 110		
	RLC (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d → 00 000 110		
	RLC (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d → 00 000 110		
	RL r		1	1	0	P	0	1	2	8	11 001 011 00 010 r ㊦		
	RL (HL)		1	1	0	P	0	1	2	15	11 001 011 00 010 110		
	RL (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d → 00 010 110		
	RL (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d → 00 010 110		
	RRC r		1	1	0	P	0	1	2	8	11 001 011 00 001 r ㊦		
	RRC (HL)		1	1	0	P	0	1	2	15	11 001 011 00 001 110		
	RRC (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d → 00 001 110		
	RRC (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d → 00 001 110		
	RR r		1	1	0	P	0	1	2	8	11 001 011 00 011 r ㊦		
	RR (HL)		1	1	0	P	0	1	2	15	11 001 011 00 011 110		
	RR(IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d → 00 011 110		
	RR (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d → 00 011 110		
	SLA r		1	1	0	P	0	1	2	8	11 001 011 00 100 r ㊦		
	SLA (HL)		1	1	0	P	0	1	2	15	11 001 011 00 100 110		
	SLA (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d → 00 100 110		
	SLA (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d → 00 100 110		



命令群	ニーモニック	オペレーション	フ ラ グ						バイト	ステート	O P コード		
			S	Z	H	P/V	N	C			76	543	210
ロー レ ジ ス タ ・ シ フ ト 命 令	SRA r	 r, (HL), (IX+d), (IY+d)	1	1	0	P	0	1	2	8	11 001 011 00 101 r	E	
	SRA (HL)		1	1	0	P	0	1	2	15	11 001 011 00 101 110		
	SRA (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d →		
	SRA (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d →		
	SRL r	 r, (HL), (IX+d), (IY+d)	1	1	0	P	0	1	2	8	11 001 011 00 111 r	E	
	SRL (HL)		1	1	0	P	0	1	2	15	11 001 011 00 111 110		
	SRL (IX+d)		1	1	0	P	0	1	4	23	11 011 101 11 001 011 ← d →		
	SRL (IY+d)		1	1	0	P	0	1	4	23	11 111 101 11 001 011 ← d →		
	RLD	A  (HL)	1	1	0	P	0	·	2	18	11 101 101 01 101 111		
	RRD	A  (HL)	1	1	0	P	0	·	2	18	11 101 101 01 100 111		
ビ ット 操 作 命 令	BIT b, r	$Z \leftarrow r_b$	×	1	1	×	0	·	2	8	11 001 011 01 b r	EF	
	BIT b, (HL)	$Z \leftarrow (HL)_b$	×	1	1	×	0	·	2	12	11 001 011 01 b 110		F
	BIT b, (IX+d)	$Z \leftarrow (IX+d)_b$	×	1	1	×	0	·	4	20	11 011 101 11 001 011 ← d →	F	
	BIT b, (IY+d)	$Z \leftarrow (IY+d)_b$	×	1	1	×	0	·	4	20	11 111 101 11 001 011 ← d →		F
	SET b, r	$r_b \leftarrow 1$	·	·	·	·	·	·	2	8	11 001 011 11 b r	EF	
	SET b, (HL)	$(HL)_b \leftarrow 1$	·	·	·	·	·	·	2	15	11 001 011 11 b 110		F
	SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	·	·	·	·	·	·	4	23	11 011 101 11 001 011 ← d →	F	
	SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	·	·	·	·	·	·	4	23	11 111 101 11 001 011 ← d →		F
	RES b, r	$r_b \leftarrow 0$	·	·	·	·	·	·	2	8	11 001 011 10 d r	EF	
	RES b, (HL)	$(HL)_b \leftarrow 0$	·	·	·	·	·	·	2	15	11 001 011 10 b 110		F
	RES b, (IX+d)	$(IX+d)_b \leftarrow 0$	·	·	·	·	·	·	4	23	11 011 101 11 001 011 ← d →	F	
	RES b, (IY+d)	$(IY+d)_b \leftarrow 0$	·	·	·	·	·	·	4	23	11 111 101 11 001 011 ← d →		F



命令群	ニーモニック	オペレーション	フ ラ グ							バイト	ステート	O P コード		
			S	Z	H	P/V	N	C	76			543	210	
ジャンプ・コール・リターン命令	JP nn	PC←nn	.	.	.	.	.	.	.	3	10	11 000 011	← n →	
	JP cc, nn	If cc is true PC←nn Otherwise continue	.	.	.	.	.	.	.	3	10	11 cc 010	← n →	Ⓒ
	JR e	PC←PC+e	.	.	.	.	.	.	.	2	12	00 011 000	← e-2 →	
	JR C, e	If C=0 continue If C=1 PC←PC+e	.	.	.	.	.	.	.	2	7 if C=0 12 if C=1	00 111 000	← e-2 →	
	JR NC, e	If C=1 continue If C=0 PC←PC+e	.	.	.	.	.	.	.	2	7 if C=1 12 if C=0	00 110 000	← e-2 →	
	JR Z, e	If Z=0 continue If Z=1 PC←PC+e	.	.	.	.	.	.	.	2	7 if Z=0 12 if Z=1	00 101 000	← e-2 →	
	JR NZ, e	If Z=1 continue If Z=0 PC←PC+e	.	.	.	.	.	.	.	2	7 if Z=1 12 if Z=0	00 100 000	← e-2 →	
	JP (HL)	PC←HL	.	.	.	.	.	.	.	1	4	11 101 001		
	JP (IX)	PC←IX	.	.	.	.	.	.	.	2	8	11 011 101		
	JP (IY)	PC←IY	.	.	.	.	.	.	.	2	8	11 111 101		
	DJNZ e	B←B-1 if B=0 continue if B≠0 PC←PC+e	.	.	.	.	.	.	.	2	8 if B=0 13 if B≠0	00 010 000	← e-2 →	
	CALL nn	(SP-1)←PC <sub>H</sub> (SP-2)←PC <sub>L</sub> PC←nn	.	.	.	.	.	.	.	3	17	11 001 101	← n →	
	CALL cc, nn	If cc is false continue otherwise same as CALL nn	.	.	.	.	.	.	.	3	10 if cc is false 17 if cc is true	11 cc 100	← n →	Ⓒ
	RET	PC <sub>L</sub> ←(SP) PC <sub>H</sub> ←(SP+1)	.	.	.	.	.	.	.	1	10	11 001 001		
	RET cc	If cc is false continue otherwise same as RET	.	.	.	.	.	.	.	1	5 if cc is false 11 if cc is true	11 cc 000		Ⓒ
	RETI	Return from interrupt	.	.	.	.	.	.	.	2	14	11 101 101	01 001 101	
	RETN	Return from non maskable interrupt	.	.	.	.	.	.	.	2	14	11 101 101	01 000 101	
	RST p	(SP-1)←PC <sub>H</sub> (SP-2)←PC <sub>L</sub> PC <sub>H</sub> ←0 PC <sub>L</sub> ←p	.	.	.	.	.	.	.	1	11	11 t 111		Ⓗ
入出力命令	IN A, n	A←(n) A <sub>0-7</sub> ←n A <sub>8-15</sub> ←A	.	.	.	.	.	.	.	2	11	11 011 011	← n →	
	IN r, (C)	r←(C) if r=110 only the flags will be affected A <sub>0-7</sub> ←C A <sub>8-15</sub> ←B	1	1	1	P	0	.	.	2	12	11 101 101	01 r 000	Ⓔ
	INI	(HL)←(C) B←B-1 HL←HL+1 A <sub>0-7</sub> ←C A <sub>8-15</sub> ←B	×	③	×	×	1	.	.	2	16	11 101 101	10 100 010	
	INIR	(HL)←(C) B←B-1 HL←HL+1 until B=0 A <sub>0-7</sub> ←C A <sub>8-15</sub> ←B	×	1	×	×	1	.	.	2	21 if B≠0 16 if B=0	11 101 101	10 110 010	
	IND	(HL)←(C) B←B-1 HL←HL-1 A <sub>0-7</sub> ←C A <sub>8-15</sub> ←B	×	③	×	×	1	.	.	2	16	11 101 101	10 101 010	



命令群	ニーモニク	オペレーション	フ ラ グ					バイト	ステート	OPコード		
			S	Z	H	P/V	N	C		76	543	210
入 出 力 命 令	INDR	(HL) ← (C) B ← B-1 HL ← HL-1 until B=0 A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	×	1	×	×	1	·	2	21 if B≠0 16 if B=0	11 101 10 111 010	101
	OUT n, A	(n) ← A A <sub>0-7</sub> ← n A <sub>8-15</sub> ← A	·	·	·	·	·	·	2	11	11 010 ← n →	011
	OUT (C), r	(C) ← r A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	·	·	·	·	·	·	2	12	11 101 01 r 001	101 ⑤
	OUTI	(C) ← (HL) B ← B-1 HL ← HL+1 A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	×	③	×	×	1	·	2	16	11 101 10 100 011	101
	OTIR	(C) ← (HL) B ← B-1 HL ← HL+1 until B=0 A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	×	1	×	×	1	·	2	21 if B≠0 16 if B=0	11 101 10 110 011	101
	OUTD	(C) ← (HL) B ← B-1 HL ← HL-1 A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	×	③	×	×	1	·	2	16	11 101 10 101 011	101
	OTDR	(C) ← (HL) B ← B-1 HL ← HL-1 until B=0 A <sub>0-7</sub> ← C A <sub>8-15</sub> ← B	×	1	×	×	1	·	2	21 if B≠0 16 if B=0	11 101 10 111 011	101

A		B		C		D		E		F		G				H	
Reg	ss dd	Reg	qq	Reg	rr	Reg	rr	Reg	r, r	Bit	b	cc	Condition	Flag	P	t	
B C	00	B C	00	B C	00	B C	00	B	000	0	000	000	N Z	Non Zero	Z	00H	000
D E	01	D E	01	D E	01	D E	01	C	001	1	001	001	Z	Zero	Z	08H	001
H L	10	H L	10	I X	10	I Y	10	D	010	2	010	010	N C	Non Carry	C	10H	010
S P	11	A F	11	S P	11	S P	11	E	011	3	011	011	C	Carry	C	18H	011
								H	100	4	100	100	P O	Parity Odd	P/V	20H	100
								L	101	5	101	101	P E	Parity Even	P/V	28H	101
								A	111	6	110	110	P	Sign Positive	S	30H	110
										7	111	111	M	Sign Negative	S	38H	111

d, n : 8ビット・イミディエト・データ  
e : 相対アドレッシングの変位置  
e-2 : eの実効変位置

#### フラグ

- : 影響受けない
  - 1 : 演算結果に従った影響を受ける
  - 0 : リセット
  - P : "1"偶数パリティ, "0"奇数パリティ
  - 1 : セット
  - V : "1"オーバフロー有り, "0"オーバフロー無し
  - ×
  - IFF : P/Vフラグ (IFF)
- ① BC-1=0 ならば P/V=0, その他 P/V=1  
② A=(HL) ならば Z=1, その他 Z=0  
③ B-1=0 ならば Z=1, その他 Z=0



# 付録B $\mu$ COM-82マシン語 $\leftrightarrow$ ニーモニック対応表

マ シ ン 語		ニ ー モ ニ ッ ク	
00 NOP		80 ADD A, B	C0 RET NZ
01 LD BC, nn		81 ADD A, C	C1 POP BC
02 LD (BC), A		82 ADD A, D	C2 JP NZ, nn
03 INC BC		83 ADD A, E	C3 JP nn
04 INC B		84 ADD A, H	C4 CALL NZ, nn
05 DEC B		85 ADD A, L	C5 PUSH BC
06 LD B, n		86 ADD A, (HL)	C6 ADD A, n
07 RLCA		87 ADD A, A	C7 RST 00H
08 EX AF, AF		88 ADC A, B	C8 RET Z
09 ADD HL, BC		89 ADC A, C	C9 RET
0A LD A, (BC)		8A ADC A, D	CA JP Z, nn
0B DEC BC		8B ADC A, E	CB <input type="text"/>
0C INC C		8C ADC A, H	CC CALL Z, nn
0D DEC C		8D ADC A, L	CD CALL nn
0E LD C, n		8E ADC A, (HL)	CE ADC A, n
0F RRCA		8F ADC A, A	CF RST 08H
10 DJNZ e		90 SUB B	D0 RET NC
11 LD DE, nn		91 SUB C	D1 POP DE
12 LD (DE), A		92 SUB D	D2 JP NC, nn
13 INC DE		93 SUB E	D3 OUT n, A
14 INC D		94 SUB H	D4 CALL NC, nn
15 DEC D		95 SUB L	D5 PUSH DE
16 LD D, n		96 SUB (HL)	D6 SUB n
17 RLA		97 SUB A	D7 RST 10H
18 JR e		98 SBC A, B	D8 RET C
19 ADD HL, DE		99 SBC A, C	D9 EXX
1A LD A, (DE)		9A SBC A, D	DA JP C, nn
1B DEC DE		9B SBC A, E	DB IN A, n
1C INC E		9C SBC A, H	DC CALL C, nn
1D DEC E		9D SBC A, L	DD <input type="text"/>
1E LD E, n		9E SBC A, (HL)	DE SBC A, n
1F RRA		9F SBC A, A	DF RST 18H
20 JR NZ, e		A0 AND B	E0 RET PO
21 LD HL, nn		A1 AND C	E1 POP HL
22 LD (nn), HL		A2 AND D	E2 JP PO, nn
23 INC HL		A3 AND E	E3 EX (SP), HL
24 INC H		A4 AND H	E4 CALL PO, nn
25 DEC H		A5 AND L	E5 PUSH HL
26 LD H, n		A6 AND (HL)	E6 AND n
27 DAA		A7 AND A	E7 RST 20H
28 JR Z, e		A8 XOR B	E8 RET PE
29 ADD HL, HL		A9 XOR C	E9 JP (HL)
2A LD HL, (nn)		AA XOR D	EA JP PE, nn
2B DEC HL		AB XOR E	EB EX DE, HL
2C INC L		AC XOR H	EC CALL PE, nn
2D DEC L		AD XOR L	ED <input type="text"/>
2E L, n		AE XOR (HL)	EE XOR n
2F CPL		AF XOR A	EF RST 28H
30 JR NC, e		B0 OR B	F0 RET P
31 LD SP, nn		B1 OR C	F1 POP AF
32 LD (nn), A		B2 OR D	F2 JP P, nn
33 INC SP		B3 OR E	F3 DI
34 INC (HL)		B4 OR H	F4 CALL P, nn
35 DEC (HL)		B5 OR L	F5 PUSH AF
36 LD (HL), n		B6 OR (HL)	F6 OR n
37 SCF		B7 OR A	F7 RST 30H
38 JR C, e		B8 CP B	F8 RET M
39 ADD HL, SP		B9 CP C	F9 LD SP, HL
3A LD A, (nn)		BA CP D	FA JP M, nn
3B DEC SP		BB CP E	FB EI
3C INC A		BC CP H	FC CALL M, nn
3D DEC A		BD CP L	FD <input type="text"/>
3E LD A, n		BE CP (HL)	FE CP n
3F CCF		BF CP A	FF RST 38H
40 LD B, B			
41 LD B, C			
42 LD B, D			
43 LD B, E			
44 LD B, H			
45 LD B, L			
46 LD B, (HL)			
47 LD B, A			
48 LD C, B			
49 LD C, C			
4A LD C, D			
4B LD C, E			
4C LD C, H			
4D LD C, L			
4E LD C, (HL)			
4F LD C, A			
50 LD D, B			
51 LD D, C			
52 LD D, D			
53 LD D, E			
54 LD D, H			
55 LD D, L			
56 LD D, (HL)			
57 LD D, A			
58 LD E, B			
59 LD E, C			
5A LD E, D			
5B LD E, E			
5C LD E, H			
5D LD E, L			
5E LD E, (HL)			
5F LD E, A			
60 LD H, B			
61 LD H, C			
62 LD H, D			
63 LD H, E			
64 LD H, H			
65 LD H, L			
66 LD H, (HL)			
67 LD H, A			
68 LD L, B			
69 LD L, C			
6A LD L, D			
6B LD L, E			
6C LD L, H			
6D LD L, L			
6E LD L, (HL)			
6F LD L, A			
70 LD (HL), B			
71 LD (HL), C			
72 LD (HL), D			
73 LD (HL), E			
74 LD (HL), H			
75 LD (HL), L			
76 HALT			
77 LD (HL), A			
78 LD A, B			
79 LD A, C			
7A LD A, D			
7B LD A, E			
7C LD A, H			
7D LD A, L			
7E LD A, (HL)			
7F LD A, A			



CB    ××											
00	RLC	B	40	BIT	0, B	80	RES	0, B	C0	SET	0, B
01	RLC	C	41	BIT	0, C	81	RES	0, C	C1	SET	0, C
02	RLC	D	42	BIT	0, D	82	RES	0, D	C2	SET	0, D
03	RLC	E	43	BIT	0, E	83	RES	0, E	C3	SET	0, E
04	RLC	H	44	BIT	0, H	84	RES	0, H	C4	SET	0, H
05	RLC	L	45	BIT	0, L	85	RES	0, L	C5	SET	0, L
06	RLC	(HL)	46	BIT	0, (HL)	86	RES	0, (HL)	C6	SET	0, (HL)
07	RLC	A	47	BIT	0, A	87	RES	0, A	C7	SET	0, A
08	RRC	B	48	BIT	1, B	88	RES	1, B	C8	SET	1, B
09	RRC	C	49	BIT	1, C	89	RES	1, C	C9	SET	1, C
0A	RRC	D	4A	BIT	1, D	8A	RES	1, D	CA	SET	1, D
0B	RRC	E	4B	BIT	1, E	8B	RES	1, E	CB	SET	1, E
0C	RRC	H	4C	BIT	1, H	8C	RES	1, H	CC	SET	1, H
0D	RRC	L	4D	BIT	1, L	8D	RES	1, L	CD	SET	1, L
0E	RRC	(HL)	4E	BIT	1, (HL)	8E	RES	1, (HL)	CE	SET	1, (HL)
0F	RRC	A	4F	BIT	1, A	8F	RES	1, A	CF	SET	1, A
10	RL	B	50	BIT	2, B	90	RES	2, B	D0	SET	2, B
11	RL	C	51	BIT	2, C	91	RES	2, C	D1	SET	2, C
12	RL	D	52	BIT	2, D	92	RES	2, D	D2	SET	2, D
13	RL	E	53	BIT	2, E	93	RES	2, E	D3	SET	2, E
14	RL	H	54	BIT	2, H	94	RES	2, H	D4	SET	2, H
15	RL	L	55	BIT	2, L	95	RES	2, L	D5	SET	2, L
16	RL	(HL)	56	BIT	2, (HL)	96	RES	2, (HL)	D6	SET	2, (HL)
17	RL	A	57	BIT	2, A	97	RES	2, A	D7	SET	2, A
18	RR	B	58	BIT	3, B	98	RES	3, B	D8	SET	3, B
19	RR	C	59	BIT	3, C	99	RES	3, C	D9	SET	3, C
1A	RR	D	5A	BIT	3, D	9A	RES	3, D	DA	SET	3, D
1B	RR	E	5B	BIT	3, E	9B	RES	3, E	DB	SET	3, E
1C	RR	H	5C	BIT	3, H	9C	RES	3, H	DC	SET	3, H
1D	RR	L	5D	BIT	3, L	9D	RES	3, L	DD	SET	3, L
1E	RR	(HL)	5E	BIT	3, (HL)	9E	RES	3, (HL)	DE	SET	3, (HL)
1F	RR	A	5F	BIT	3, A	9F	RES	3, A	DF	SET	3, A
20	SLA	B	60	BIT	4, B	A0	RES	4, B	E0	SET	4, B
21	SLA	C	61	BIT	4, C	A1	RES	4, C	E1	SET	4, C
22	SLA	D	62	BIT	4, D	A2	RES	4, D	E2	SET	4, D
23	SLA	E	63	BIT	4, E	A3	RES	4, E	E3	SET	4, E
24	SLA	H	64	BIT	4, H	A4	RES	4, H	E4	SET	4, H
25	SLA	L	65	BIT	4, L	A5	RES	4, L	E5	SET	4, L
26	SLA	(HL)	66	BIT	4, (HL)	A6	RES	4, (HL)	E6	SET	4, (HL)
27	SLA	A	67	BIT	4, A	A7	RES	4, A	E7	SET	4, A
28	SRA	B	68	BIT	5, B	A8	RES	5, B	E8	SET	5, B
29	SRA	C	69	BIT	5, C	A9	RES	5, C	E9	SET	5, C
2A	SRA	D	6A	BIT	5, D	AA	RES	5, D	EA	SET	5, D
2B	SRA	E	6B	BIT	5, E	AB	RES	5, E	EB	SET	5, E
2C	SRA	H	6C	BIT	5, H	AC	RES	5, H	EC	SET	5, H
2D	SRA	L	6D	BIT	5, L	AD	RES	5, L	ED	SET	5, L
2E	SRA	(HL)	6E	BIT	5, (HL)	AE	RES	5, (HL)	EE	SET	5, (HL)
2F	SRA	A	6F	BIT	5, A	AF	RES	5, A	EF	SET	5, A
30			70	BIT	6, B	B0	RES	6, B	F0	SET	6, B
31			71	BIT	6, C	B1	RES	6, C	F1	SET	6, C
32			72	BIT	6, D	B2	RES	6, D	F2	SET	6, D
33			73	BIT	6, E	B3	RES	6, E	F3	SET	6, E
34			74	BIT	6, H	B4	RES	6, H	F4	SET	6, H
35			75	BIT	6, L	B5	RES	6, L	F5	SET	6, L
36			76	BIT	6, (HL)	B6	RES	6, (HL)	F6	SET	6, (HL)
37			77	BIT	6, A	B7	RES	6, A	F7	SET	6, A
38	SRL	B	78	BIT	7, B	B8	RES	7, B	F8	SET	7, B
39	SRL	C	79	BIT	7, C	B9	RES	7, C	F9	SET	7, C
3A	SRL	D	7A	BIT	7, D	BA	RES	7, D	FA	SET	7, D
3B	SRL	E	7B	BIT	7, E	BB	RES	7, E	FB	SET	7, E
3C	SRL	H	7C	BIT	7, H	BC	RES	7, H	FC	SET	7, H
3D	SRL	L	7D	BIT	7, L	BD	RES	7, L	FD	SET	7, L
3E	SRL	(HL)	7E	BIT	7, (HL)	BE	RES	7, (HL)	FE	SET	7, (HL)
3F	SRL	A	7F	BIT	7, A	BF	RES	7, A	FF	SET	7, A



D D × ×				E D × ×				F D × ×			
0 9		ADD	IX, BC	4 0	IN	B, (C)		0 9		ADD	IY, BC
1 9		ADD	IX, DE	4 1	OUT	(C), B		1 9		ADD	IY, DE
2 1		LD	IX, nn	4 2	SBC	HL, BC		2 1		LD	IY, nn
2 2		LD	(nn), IX	4 3	LD	(nn), BC		2 2		LD	(nn), IY
2 3		INC	IX	4 4	NEG			2 3		INC	IY
2 9		ADD	IX, IX	4 5	RET N			2 9		ADD	IY, IY
2 A		LD	IX, (nn)	4 6	IM	0		2 A		LD	IY, (nn)
2 B		DEC	IX	4 7	LD	I, A		2 B		DEC	IY
3 4		INC	(IX+d)	4 8	IN	C, (C)		3 4		INC	(IY+d)
3 5		DEC	(IX+d)	4 9	OUT	(C), C		3 5		DEC	(IY+d)
3 6		LD	(IX+d), n	4 A	ADC	HL, BC		3 6		LD	(IY+d), n
3 9		ADD	IX, SP	4 B	LD	BC, (nn)		3 9		ADD	IY, SP
4 6		LD	B, (IX+d)	4 D	RET I			4 6		LD	B, (IY+d)
4 E		LD	C, (IX+d)	4 F	LD	R, A		4 E		LD	C, (IY+d)
5 6		LD	D, (IX+d)	5 0	IN	D, (C)		5 6		LD	D, (IY+d)
5 E		LD	E, (IX+d)	5 1	OUT	(C), D		5 E		LD	E, (IY+d)
6 6		LD	H, (IX+d)	5 2	SBC	HL, DE		6 6		LD	H, (IY+d)
6 E		LD	L, (IX+d)	5 3	LD	(nn), DE		6 E		LD	L, (IY+d)
7 0		LD	(IX+d), B	5 6	IN	1		7 0		LD	(IY+d), B
7 1		LD	(IX+d), C	5 7	LD	A, I		7 1		LD	(IY+d), C
7 2		LD	(IX+d), D	5 8	IN	E, (C)		7 2		LD	(IY+d), D
7 3		LD	(IX+d), E	5 9	OUT	(C), E		7 3		LD	(IY+d), E
7 4		LD	(IX+d), H	5 A	ADC	HL, DE		7 4		LD	(IY+d), H
7 5		LD	(IX+d), L	5 B	LD	DE, (nn)		7 5		LD	(IY+d), L
7 7		LD	(IX+d), A	5 E	IM	2		7 7		LD	(IY+d), A
7 E		LD	A, (IX+d)	5 F	LD	A, R		7 E		LD	A, (IY+d)
8 6		ADD	A, (IX+d)	6 0	IN	H, (C)		8 6		ADD	A, (IY+d)
8 E		ADC	A, (IX+d)	6 1	OUT	(C), H		8 E		ADC	A, (IY+d)
9 6		SUB	(IX+d)	6 2	SBC	HL, HL		9 6		SUB	(IY+d)
9 E		SBC	A, (IX+d)	6 7	RRD			9 E		SBC	A, (IY+d)
A 6		AND	(IX+d)	6 8	IN	L, (C)		A 6		AND	(IY+d)
A E		XOR	(IX+d)	6 9	OUT	(C), L		A E		XOR	(IY+d)
B 6		OR	(IX+d)	6 A	ADC	HL, HL		B 6		OR	(IY+d)
B E		CP	(IX+d)	6 F	RLD			B E		CP	(IY+d)
CB d 0 6		RLC	(IX+d)	7 2	SBC	HL, SP		CB d 0 6		RLC	(IY+d)
CB d 0 E		RRC	(IX+d)	7 3	LD	(nn), SP		CB d 0 E		RRC	(IY+d)
CB d 1 6		RL	(IX+d)	7 8	IN	A, (C)		CB d 1 6		RL	(IY+d)
CB d 1 E		RR	(IX+d)	7 9	OUT	(C), A		CB d 1 E		RR	(IY+d)
CB d 2 6		SLA	(IX+d)	7 A	ADC	HL, SP		CB d 2 6		SLA	(IY+d)
CB d 2 E		SRA	(IX+d)	7 B	LD	SP, (nn)		CB d 2 E		SRA	(IY+d)
CB d 3 E		SRL	(IX+d)	A 0	LD I			CB d 3 E		SRL	(IY+d)
CB d 4 6		BIT	0, (IX+d)	A 1	CPI			CB d 4 6		BIT	0, (IY+d)
CB d 4 E		BIT	1, (IX+d)	A 2	INI			CB d 4 E		BIT	1, (IY+d)
CB d 5 6		BIT	2, (IX+d)	A 3	OUT I			CB d 5 6		BIT	2, (IY+d)
CB d 5 E		BIT	3, (IX+d)	A 8	LDD			CB d 5 E		BIT	3, (IY+d)
CB d 6 6		BIT	4, (IX+d)	A 9	CPD			CB d 6 6		BIT	4, (IY+d)
CB d 6 E		BIT	5, (IX+d)	AA	IND			CB d 6 E		BIT	5, (IY+d)
CB d 7 6		BIT	6, (IX+d)	AB	OUTD			CB d 7 6		BIT	6, (IY+d)
CB d 7 E		BIT	7, (IX+d)	B 0	LD I R			CB d 7 E		BIT	7, (IY+d)
CB d 8 6		RES	0, (IX+d)	B 1	CPI R			CB d 8 6		RES	0, (IY+d)
CB d 8 E		RES	1, (IX+d)	B 2	INI R			CB d 8 E		RES	1, (IY+d)
CB d 9 6		RES	2, (IX+d)	B 3	OT I R			CB d 9 6		RES	2, (IY+d)
CB d 9 E		RES	3, (IX+d)	B 8	LDDR			CB d 9 E		RES	3, (IY+d)
CB d A 6		RES	4, (IX+d)	B 9	CPDR			CB d A 6		RES	4, (IY+d)
CB d A E		RES	5, (IX+d)	BA	INDR			CB d A E		RES	5, (IY+d)
CB d B 6		RES	6, (IX+d)	BB	OTDR			CB d B 6		RES	6, (IY+d)
CB d B E		RES	7, (IX+d)					CB d B E		RES	7, (IY+d)
CB d C 6		SET	0, (IX+d)					CB d C 6		SET	0, (IY+d)
CB d C E		SET	1, (IX+d)					CB d C E		SET	1, (IY+d)
CB d D 6		SET	2, (IX+d)					CB d D 6		SET	2, (IY+d)
CB d D E		SET	3, (IX+d)					CB d D E		SET	3, (IY+d)
CB d E 6		SET	4, (IX+d)					CB d E 6		SET	4, (IY+d)
CB d E E		SET	5, (IX+d)					CB d E E		SET	5, (IY+d)
CB d F 6		SET	6, (IX+d)					CB d F 6		SET	6, (IY+d)
CB d F E		SET	7, (IX+d)					CB d F E		SET	7, (IY+d)
E 1		POP	IX					E 1		POP	IY
E 3		EX	(SP), IX					E 3		EX	(SP), IY
E 5		PUSH	IX					E 5		PUSH	IY
E 9		JP	(IX)					E 9		JP	(IY)
F 9		LD	SP, IX					F 9		LD	SP, IY



# 付録C $\mu$ COM-82ニーモニック $\longleftrightarrow$ マシン語 対照表

8ビット・ロード

$\times$	I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX+d)	(IY+d)	(nn)	n
LD A, $\times$	ED 5 7	ED 5 F	7 F	7 8	7 9	7 A	7 B	7 C	7 D	7 E	0 A	1 A	DD 7 E d	FD 7 E d	3 A n	3 E n
LD B, $\times$			4 7	4 0	4 1	4 2	4 3	4 4	4 5	4 6			DD 4 6 d	FD 4 6 d		0 6 n
LD C, $\times$			4 F	4 8	4 9	4 A	4 B	4 C	4 D	4 E			DD 4 E d	FD 4 E d		0 E n
LD D, $\times$			5 7	5 0	5 1	5 2	5 3	5 4	5 5	5 6			DD 5 6 d	FD 5 6 d		1 6 n
LD E, $\times$			5 F	5 8	5 9	5 A	5 B	5 C	5 D	5 E			DD 5 E d	FD 5 E d		1 E n
LD H, $\times$			6 7	6 0	6 1	6 2	6 3	6 4	6 5	6 6			DD 6 6 d	FD 6 6 d		2 6 n
LD L, $\times$			6 F	6 8	6 9	6 A	6 B	6 C	6 D	6 E			DD 6 E d	FD 6 E d		2 E n
LD (HL), $\times$			7 7	7 0	7 1	7 2	7 3	7 4	7 5							3 6 n
LD (BC), $\times$			0 2													
LD (DE), $\times$			1 2													
LD (IX+d), $\times$			DD 7 7 d	DD 7 0 d	DD 7 1 d	DD 7 2 d	DD 7 3 d	DD 7 4 d	DD 7 5 d							DD 3 6 d n
LD (IY+d), $\times$			FD 7 7 d	FD 7 0 d	FD 7 1 d	FD 7 2 d	FD 7 3 d	FD 7 4 d	FD 7 5 d							FD 3 6 d n
LD (nn), $\times$			3 2 n n													
LD I, $\times$			ED 4 7													
LD R, $\times$			ED 4 F													

16ビット・ロード

$\times$	AF	BC	DE	HL	SP	IX	IY	nn	(nn)
LD AF, $\times$									
LD BC, $\times$								0 1 n n	ED 4 B n n
LD DE, $\times$								1 1 n n	ED 5 B n n
LD HL, $\times$								2 1 n n	2 A n n
LD SP, $\times$				F 9		DD F 9	FD F 9	3 1 n n	ED 7 B n n
LD IX, $\times$								DD 2 1 n n	DD 2 A n n
LD IY, $\times$								FD 2 1 n n	FD 2 A n n
LD (nn), $\times$		ED 4 3 n n	ED 5 3 n n	2 2 n n	ED 7 3 n n	DD 2 2 n n	FD 2 2 n n		
PUSH $\times$	F 5	C 5	D 5	E 5		DD E 5	FD E 5		
POP $\times$	F 1	C 1	D 1	E 1		DD E 1	FD E 1		

ブロック転送

LDI	ED A 0
LDIR	ED B 0
LDD	ED A 8
LDDR	ED B 8

ブロック・サーチ

CPI	ED A 1
CPIR	ED B 1
CPD	ED A 9
CPDR	ED B 9



# 8ビット算術論理演算

×	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)	n
ADD A, ×	8 7	8 0	8 1	8 2	8 3	8 4	8 5	8 6	DD 8 6 d	FD 8 6 d	C 6 n
ADC A, ×	8 F	8 8	8 9	8 A	8 B	8 C	8 D	8 E	DD 8 E d	FD 8 E d	C E n
SUB ×	9 7	9 0	9 1	9 2	9 3	9 4	9 5	9 6	DD 9 6 d	FD 9 6 d	D 6 n
SBC A, ×	9 F	9 8	9 9	9 A	9 B	9 C	9 D	9 E	DD 9 E d	FD 9 E d	D E n
AND ×	A 7	A 0	A 1	A 2	A 3	A 4	A 5	A 6	DD A 6 d	FD A 6 d	E 6 n
XOR ×	A F	A 8	A 9	A A	A B	A C	A D	A E	DD A E d	FD A E d	E E n
OR ×	B 7	B 0	B 1	B 2	B 3	B 4	B 5	B 6	DD B 6 d	FD B 6 d	F 6 n
CP ×	B F	B 8	B 9	B A	B B	B C	B D	B E	DD B E d	FD B E d	F E n
INC ×	3 C	0 4	0 C	1 4	1 C	2 4	2 C	3 4	DD 3 4 d	FD 3 4 d	
DEC ×	3 D	0 5	0 D	1 5	1 D	2 5	2 D	3 5	DD 3 5 d	FD 3 5 d	

# CPUコントロール

NOP	0 0
HALT	7 6
DI	F 3
EI	F B
IM 0	E D 4 6
IM 1	E D 5 6
IM 2	E D 5 E

# 16ビット算術演算

×	BC	DE	HL	SP	IX	IY
ADD HL, ×	0 9	1 9	2 9	3 9		
ADD IX, ×	DD 0 9	DD 1 9		DD 3 9	DD 2 9	
ADD IY, ×	FD 0 9	FD 1 9		FD 3 9		FD 2 9
ADC HL, ×	ED 4 A	ED 5 A	ED 6 A	ED 7 A		
SBC HL, ×	ED 4 2	ED 5 2	ED 6 2	ED 7 2		
INC ×	0 3	1 3	2 3	3 3	DD 2 3	FD 2 3
DEC ×	0 B	1 B	2 B	3 B	DD 2 B	FD 2 B

# エクスチェンジ

EX AF, AF'	0 8
EX DE, HL	E B
EX (SP), HL	E 3
EX (SP), IX	DD E 3
EX (SP), IY	FD E 3
EXX	D 9

# アキュムレータ操作

DAA	2 7
CPL	2 F
NEG	E D 4 4
CCF	3 F
SCF	3 7



ローテート、シフト

×	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)
RLC ×	CB 0 7	CB 0 0	CB 0 1	CB 0 2	CB 0 3	CB 0 4	CB 0 5	CB 0 6	DD CB d 0 6	FD CB d 0 6
RRC ×	CB 0 F	CB 0 8	CB 0 9	CB 0 A	CB 0 B	CB 0 C	CB 0 D	CB 0 E	DD CB d 0 E	FD CB d 0 E
RL ×	CB 1 7	CB 1 0	CB 1 1	CB 1 2	CB 1 3	CB 1 4	CB 1 5	CB 1 6	DD CB d 1 6	FD CB d 1 6
RR ×	CB 1 F	CB 1 8	CB 1 9	CB 1 A	CB 1 B	CB 1 C	CB 1 D	CB 1 E	DD CB d 1 E	FD CB d 1 E
SLA ×	CB 2 7	CB 2 0	CB 2 1	CB 2 2	CB 2 3	CB 2 4	CB 2 5	CB 2 6	DD CB d 2 6	FD CB d 2 6
SRA ×	CB 2 F	CB 2 8	CB 2 9	CB 2 A	CB 2 B	CB 2 C	CB 2 D	CB 2 E	DD CB d 2 E	FD CB d 2 E
SRL ×	CB 3 F	CB 3 8	CB 3 9	CB 3 A	CB 3 B	CB 3 C	CB 3 D	CB 3 E	DD CB d 3 E	FD CB d 3 E
RLD								ED 6 F		
RRD								ED 6 7		

	A
RLCA	0 7
RRCA	0 F
RLA	1 7
RRA	1 F

ジャンプ、コール、リターン

×	UN COND	C	NC	Z	NZ	PE	PO	M	P	
JP ×, nn	C 3 n n	D A n n	D 2 n n	C A n n	C 2 n n	E A n n	E 2 n n	F A n n	F 2 n n	
JR ×, e	1 8 e-2	3 8 e-2	3 0 e-2	2 8 e-2	2 0 e-2					
JP (HL)	E 9									
JP (IX)	DD E 9									
JP (IY)	FD E 9									
CALL ×, nn	CD n n	DC n n	D 4 n n	CC n n	C 4 n n	EC n n	E 4 n n	FC n n	F 4 n n	
DJNZ e										1 0 e-2
RET ×	C 9	D 8	D 0	C 8	C 0	E 8	E 0	F 8	F 0	
RETI	ED 4 D									
RETN	ED 4 5									

リスタート

RST 00H	C 7
RST 08H	C F
RST 10H	D 7
RST 18H	D F
RST 20H	E 7
RST 28H	E F
RST 30H	F 7
RST 38H	F F



ビット操作

×	A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)
BIT 0, ×	CB 4 7	CB 4 0	CB 4 1	CB 4 2	CB 4 3	CB 4 4	CB 4 5	CB 4 6	DD CB d 4 6	FD CB d 4 6
BIT 1, ×	CB 4 F	CB 4 8	CB 4 9	CB 4 A	CB 4 B	CB 4 C	CB 4 D	CB 4 E	DD CB d 4 E	FD CB d 4 E
BIT 2, ×	CB 5 7	CB 5 0	CB 5 1	CB 5 2	CB 5 3	CB 5 4	CB 5 5	CB 5 6	DD CB d 5 6	FD CB d 5 6
BIT 3, ×	CB 5 F	CB 5 8	CB 5 9	CB 5 A	CB 5 B	CB 5 C	CB 5 D	CB 5 E	DD CB d 5 E	FD CB d 5 E
BIT 4, ×	CB 6 7	CB 6 0	CB 6 1	CB 6 2	CB 6 3	CB 6 4	CB 6 5	CB 6 6	DD CB d 6 6	FD CB d 6 6
BIT 5, ×	CB 6 F	CB 6 8	CB 6 9	CB 6 A	CB 6 B	CB 6 C	CB 6 D	CB 6 E	DD CB d 6 E	FD CB d 6 E
BIT 6, ×	CB 7 7	CB 7 0	CB 7 1	CB 7 2	CB 7 3	CB 7 4	CB 7 5	CB 7 6	DD CB d 7 6	FD CB d 7 6
BIT 7, ×	CB 7 F	CB 7 8	CB 7 9	CB 7 A	CB 7 B	CB 7 C	CB 7 D	CB 7 E	DD CB d 7 E	FD CB d 7 E
RES 0, ×	CB 8 7	CB 8 0	CB 8 1	CB 8 2	CB 8 3	CB 8 4	CB 8 5	CB 8 6	DD CB d 8 6	FD CB d 8 6
RES 1, ×	CB 8 F	CB 8 8	CB 8 9	CB 8 A	CB 8 B	CB 8 C	CB 8 D	CB 8 E	DD CB d 8 E	FD CB d 8 E
RES 2, ×	CB 9 7	CB 9 0	CB 9 1	CB 9 2	CB 9 3	CB 9 4	CB 9 5	CB 9 6	DD CB d 9 6	FD CB d 9 6
RES 3, ×	CB 9 F	CB 9 8	CB 9 9	CB 9 A	CB 9 B	CB 9 C	CB 9 D	CB 9 E	DD CB d 9 E	FD CB d 9 E
RES 4, ×	CB A 7	CB A 0	CB A 1	CB A 2	CB A 3	CB A 4	CB A 5	CB A 6	DD CB d A 6	FD CB d A 6
RES 5, ×	CB A F	CB A 8	CB A 9	CB A A	CB A B	CB A C	CB A D	CB A E	DD CB d A E	FD CB d A E
RES 6, ×	CB B 7	CB B 0	CB B 1	CB B 2	CB B 3	CB B 4	CB B 5	CB B 6	DD CB d B 6	FD CB d B 6
RES 7, ×	CB B F	CB B 8	CB B 9	CB B A	CB B B	CB B C	CB B D	CB B E	DD CB d B E	FD CB d B E
SET 0, ×	CB C 7	CB C 0	CB C 1	CB C 2	CB C 3	CB C 4	CB C 5	CB C 6	DD CB d C 6	FD CB d C 6
SET 1, ×	CB C F	CB C 8	CB C 9	CB C A	CB C B	CB C C	CB C D	CB C E	DD CB d C E	FD CB d C E
SET 2, ×	CB D 7	CB D 0	CB D 1	CB D 2	CB D 3	CB D 4	CB D 5	CB D 6	DD CB d D 6	FD CB d D 6
SET 3, ×	CB D F	CB D 8	CB D 9	CB D A	CB D B	CB D C	CB D D	CB D E	DD CB d D E	FD CB d D E
SET 4, ×	CB E 7	CB E 0	CB E 1	CB E 2	CB E 3	CB E 4	CB E 5	CB E 6	DD CB d E 6	FD CB d E 6
SET 5, ×	CB E F	CB E 8	CB E 9	CB E A	CB E B	CB E C	CB E D	CB E E	DD CB d E E	FD CB d E E
SET 6, ×	CB F 7	CB F 0	CB F 1	CB F 2	CB F 3	CB F 4	CB F 5	CB F 6	DD CB d F 6	FD CB d F 6
SET 7, ×	CB F F	CB F 8	CB F 9	CB F A	CB F B	CB F C	CB F D	CB F E	DD CB d F E	FD CB d F E

入 力

IN A, n	DB n
IN A, (C)	ED 7 8
IN B, (C)	ED 4 0
IN C, (C)	ED 4 8
IN D, (C)	ED 5 0
IN E, (C)	ED 5 8
IN H, (C)	ED 6 0
IN L, (C)	ED 6 8
INI	ED A 2
INIR	ED B 2
IND	ED A A
INDR	ED B A

出 力

OUT n, A	D 3 n
OUT (C), A	ED 7 9
OUT (C), B	ED 4 1
OUT (C), C	ED 4 9
OUT (C), D	ED 5 1
OUT (C), E	ED 5 9
OUT (C), H	ED 6 1
OUT (C), L	ED 6 9
OUTI	ED A 3
OTIR	ED B 3
OUTD	ED A B
OTDR	ED B B



# 付録D 10進 $\longleftrightarrow$ 16進変換表

下位 上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	62	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

最上位× 桁	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
× 0 0	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840
× 0 0 0	4096	8192	12288	16384	20480	24576	28672	32768	36864	40960	45056	49152	53248	57344	61440
× 0 0 0 0	65536														



# 付録E 1バイト符号付16進数

下位 上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	-128	-127	-126	-125	-124	-123	-122	-121	-120	-119	-118	-117	-116	-115	-114	-113
9	-112	-111	-110	-109	-108	-107	-106	-105	-104	-103	-102	-101	-100	-99	-98	-97
A	-96	-95	-94	-93	-92	-91	-90	-89	-88	-87	-86	-85	-84	-83	-82	-81
B	-80	-79	-78	-77	-76	-75	-74	-73	-72	-71	-70	-69	-68	-67	-66	-65
C	-64	-63	-62	-61	-60	-59	-58	-57	-56	-55	-54	-53	-52	-51	-50	-49
D	-48	-47	-46	-45	-44	-43	-42	-41	-40	-39	-38	-37	-36	-35	-34	-33
E	-32	-31	-30	-29	-28	-27	-26	-25	-24	-23	-22	-21	-20	-19	-18	-17
F	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1



# 侵入者

プログラミングをしていると、突然ハッカーが侵入して来た。

もちろん、私がプログラムを作っているこのコンピュータには通信回線はついていない。しかし、近頃のハッカーは通信回線などなくても侵入してしまうのである。

よくはわからんが、強力な電磁波により、外部からコンタクトするらしい。実に恐ろしい奴らだ。

以下はディスプレイに現れた彼らの会話を文章化したものである。

A：「はい元気かい、ゲーム屋のオッサン。こんな時間まで、ご精がでますねえ」

B：「なんだなんだ、人がのぞき込もうとしたら、ちゃっかり先回りしている奴がいるなア」

A：「おや、お前さんものぞきに來たの？ ヒマだね、こんなゲーム作りのコンピュータに入ってくるなんてよ」

B：「ちょっとからかってやろうと思ったまでよ、お互い様だ。そっちはどこから侵入しているんだ？」

A：「さあね、地球外かもしれないぜ。ま、ハッカーには居所も名前も関係ないさ。それより、このプログラム、随分と出来てるみたいだぜ。遊んでみないか？」

B：「待ってくれ、今リストを解析中だが…… うん、チェスだぜ、こりゃあ」

A：「ほう、もうわかるとはオタクもするどい。まだ思考ルーチンは完成してないようだけど、動くみたいだ」

B：「じゃ、いっちょやってみないか？ 相手の顔も知らないハッカー同志が、他人の機械でチェスやるなんてサイコーじゃん」

ここまで、私はただポカンとディスプレイをながめていただけだった。が、我に返って、こいつは腹を立てるべき事であると気がついた。すでに苦勞して作った画面上のリストは、彼らの会話でぐしゃぐしゃになっている。

私：「こらっ、お前ら人のコンピュータで勝手な事するな！」

急いでメッセージをたたき込む。

A：「おや、こいつ何か言ってるぜ」

B：「気にするなよ、どーせ遊びじゃないの。ハッカーに逆らうとコワイのよ。大事なデータ消されたくないでしょ。だったらしばらく黙ってな。さて、実行可能ファイルを読み込むぜ」

と、ディスクランプが灯り、すぐにグラフィック画面にチェス盤が現れた。私が作っている途中のものである。

A：「フーン。グラフィックはなかなかいいんじゃない？」

B：「フルカラーのチェス盤てのは変な感じだが、ま、いいか。じゃ、どうぞお先に…」

A：「うん。それじゃ行くぜ。オレ、どっちかと言うとリアルタイムゲームの方が好きなんだけどな」

文字を表示しているテキスト画面の下にグラフィックがあるので、かなり会話が読みづらくなった。



B：「ほう、もう“キャスリング”するの。ふんふん、まあわりといい線いってるんじゃないの、君」

動いている。プログラムは作ってあるものの、このゲームが動くのを見るのは始めてだ。私は再びボーゼンとなった。

B：「それじゃあ、こっちのナイトを動かすとするか。—やや、なんだこのナイトは！」

黒のナイトは一度ケイマ跳びをしたが、それで終らず、ピョンピョンと盤面を跳び続ける。

A：「あれあれ、しょうがねえなあ、バグがいるよ。どこだ？ いたいた、直してやるか…… じゃ、また始めからだ」

B：「ちょっと待て、まだバグがあるぞ。ほら、こうやってクイーンでポーンに当てるだろう……」

黒のクイーンが味方のポーンに衝突すると、ポーンは勢いよく前にはじかれ、敵のビショップに当たった。

当てられたビショップもまたはじかれ、盤面から飛び出す。と、思いきや、ふちに当たって跳ね返り、ルークを跳ね飛ばした。ルークもまた盤のふちに当たると跳ね返って……。

見る間に盤面中の駒が跳ね回る様になった。しかも、それらはぶつかり合う度にキャラクターが壊れて行く。

駒は次第に形を変え、まるで“えへん虫”のようになってしまった。

A：「おっ、おい、やばいぜこいつは——」

B：「バグが正体を表しやがったか！ こんなバグの多いプログラムは始めてだ」

A：「そんな、ナマやさしいものじゃない。見ろよ、だんだんキャラクターの数が増えているだろう。俺はウワサに聞いた事があるんだ。こいつはコンピュータに巣食う“バグ”だぜ。ここはバグにたたられている。早くコンタクトを切らないと我々に移るぞ！」

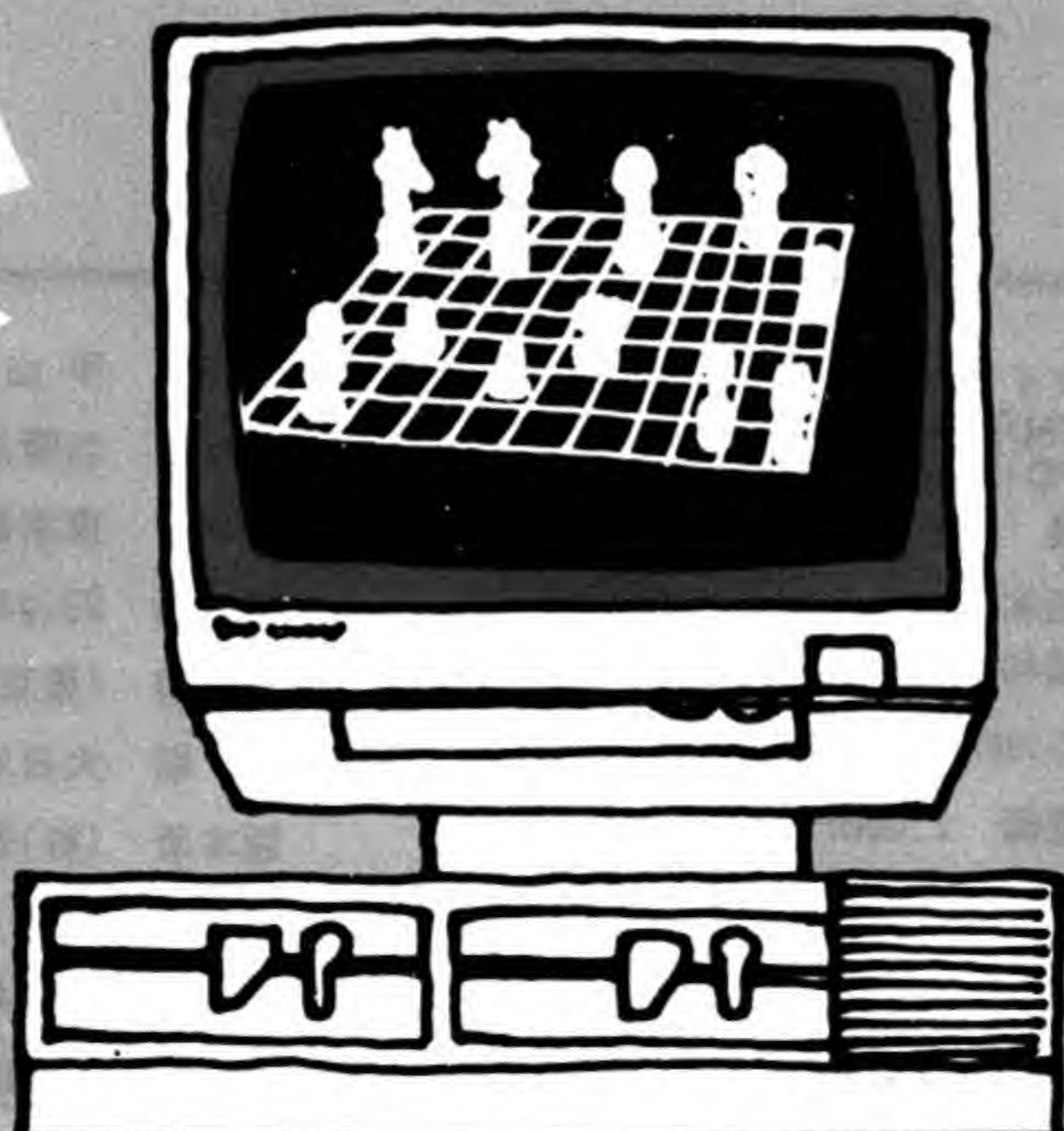
B：「馬鹿な事を言うな。……い、いや、ほ、本当にこっちへ入り込んで来たぞ。ひゃーっ！ どうすればいいんだ!？」

A：「俺の知った事か！ 俺はコンタクトを切るぜ。あばよ」

B：「お、俺も！——」

画面がすっ飛んで、リセット状態になった。

何が起こったのか、私には見当もつかない。しかし、その日以来、私のプログラムにバグは出ない。





マイコンテクニカルブックス  
FH/MHシリーズ対応  
PC-8801mkII SR/FR/MR活用研究  
呉 英二／石田 和久 共著

昭和62年 3 月26日初版発行  
昭和62年11月30日第2刷発行

© 1987 Printed in Japan

定価 2,000円 (送料300円)

発行人 平山 秀雄

発行所 株式会社電波新聞社

〒141 東京都品川区東五反田1-11-15

電 話 03(445)6111(大代表)

振 替 (東京)5-51961

印 刷 大日本印刷 株式会社

製本所 (株)堅 省 堂

〈本誌記事、プログラムの無断使用を禁止します〉

乱丁、落丁本はお取り替え致します。

ISBN4-88554-111-5 C3055 ¥2000E















**MICOM  
TECHNICAL  
BOOKS**

ISBN4-88554-111-5 C3055 ¥2000E

定価2,000円